7/95

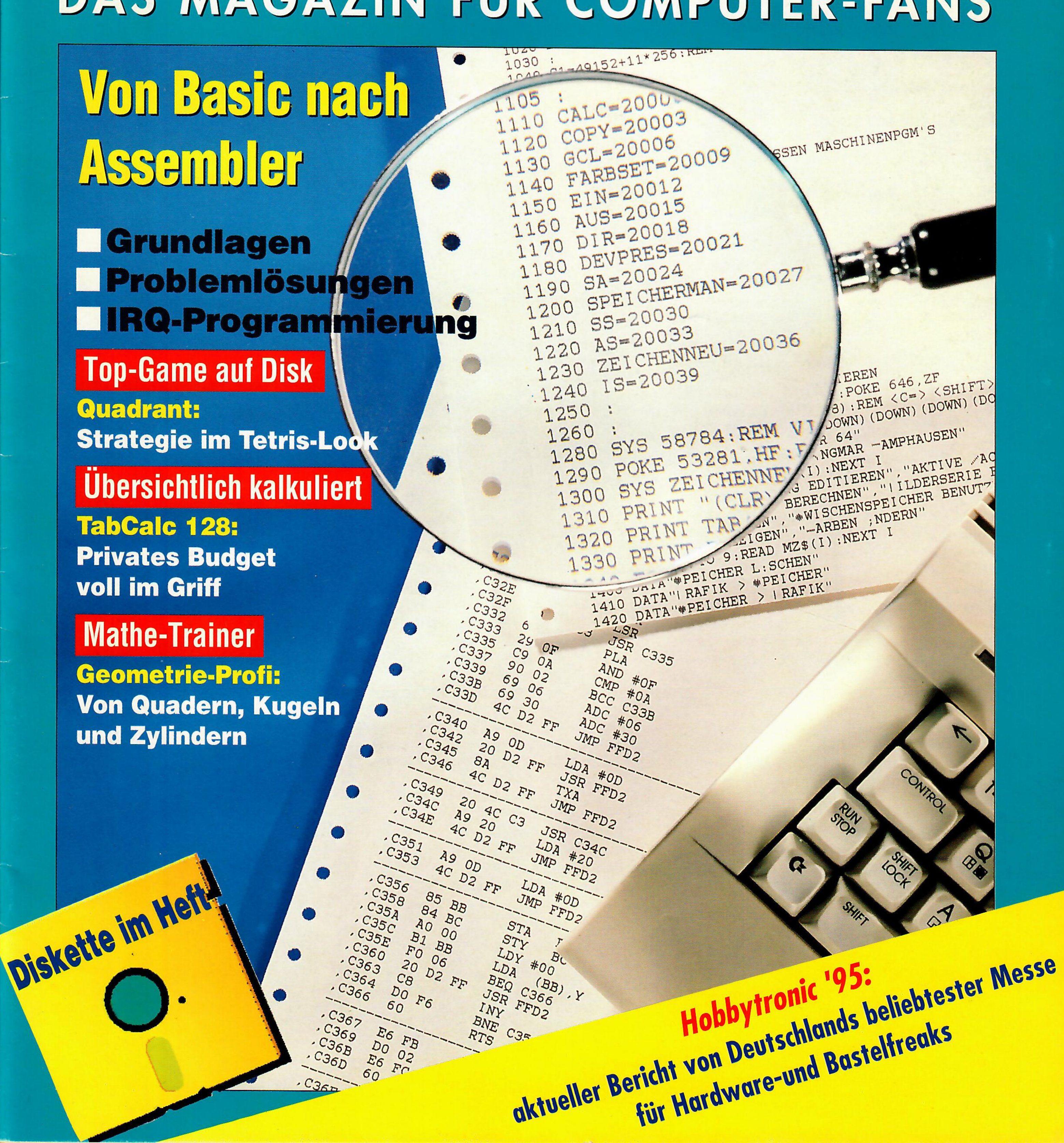
MAGNA MEDIA

öS 80,- sfr 9,80 DM 9,80

Die Nummer 1 für C64 und C128

64'er

DAS MAGAZIN FÜR COMPUTER-FANS



INHALT 7/95



Hobbytronik ´95: Das Mekka für Homecomputer- und Bastel-Fans öffnete am 10. Mai 1995 seine Tore – wir stellen die Neuheiten vor!



Aktuell

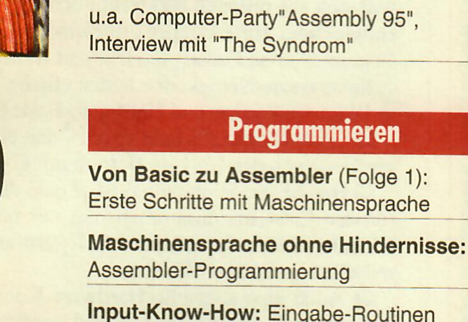
News & Facts 4

Hobbytronik 95: Messe-Streifzug 6

Scanntroniks Party: Bericht von der Jubiläumsfeier 7

Btx-Extra: Decoder für C 64/C 128 8

Szene inside: u.a. Computer-Party"Assembly 95", Interview mit "The Syndrom" 10



Programmieren

Von Basic zu Assembler (Folge 1): Erste Schritte mit Maschinensprache 14

Maschinensprache ohne Hindernisse: Assembler-Programmierung 22

Input-Know-How: Eingabe-Routinen zu Tastatur, Maus und Joystick 24

Die Sache mit dem IRQ: Geheimnisse des Rasterzeilen-Interrupts 27



Szene inside: Der "C-64-Underground" unter der Lupe - diesmal u.a. mit News zur "Assembly 95" und dem 64'er-Szene-Interview

10



Tips & Tricks

... zum C 64: u.a. synthetische Steuerzeichen, LIST-Schutz 30

... zum C 128: Zeichensatz-Editor "Ed´Char 128" mit VDC-Lader 31

... zum Plus/4: Trick-Parade mit "Directory-Auswahl" u.a. 32

Nichts geht über Maschinensprache: Quicksort 64/128 in Assembler 33

Anwendung

Zahlen am laufenden Band: TabCalc 128 V3.2 34

Der Geometrie-Profi: Mathematik zum Anfassen 35

Spiel

Quadrant: Top-Strategie-Game mit Tetris-Feeling 36



Dateiverwaltung

Datenbank GmbH (Folge 3): Druck- und Ausgabe-Routinen 38



Geos

Geos intern (Folge 6): Kernel-Routinen 40

Geos zum Anfassen (Folge 5): GeoProgrammer-Kurs 44

Neues von Geos: u.a. GeoFunktion 46

Geos-Software im Test: CLI 3.0 47

Grafik

GoDot greift nach Mega-Bytes! CMD-Laufwerke unter GoDot 48

Rubriken

Kolumne 4

Software-Klassiker auf Disk: 64´er-Assembler-Paket 12

Diskettenseite 19

Kleinanzeigenauftrag 20

Impressum 20

Computermarkt 21

Leserforum 49

Vorschau 64`er 8/95 50

Inserentenverzeichnis 50



Quadrant: Die Mischung aus "Tetris" und "Vier gewinnt", kettet den Knobelfreak mit seiner gelungen Mixtur aus Grafik, Sound und Spielspaß schnell an den Joystick!

36

Seite 14

Seite 36

Seite 34

Seite 35



Dieses Symbol zeigt an, welche Programme auf Diskette erhältlich sind

Keine Angst vor Maschinensprache!

Als ich vor beinahe zehn Jahren mit der Computerei begann, stand Basic auf der Tages-Ordnung. Als Student schrieb ich Programme zur Getriebe-Berechnung und anderen langweiligen Kram. Doch recht bald kam der Gedanke auf, daß der C 64 mehr kann, als langweilige Berechnungen und Kalkulationen auszuführen. Schnell rückten andere Projekte ins Blickfeld, die immer komplexer und umfangreicher wurden. Schon recht bald zeigten sich die Grenzen der Programmiersprache Basic und dem C 64 drohte die Luft auszugehen. Zu diesem Zeitpunkt hieß das Wunderwort: Assembler! Aber bis die ersten Maschinensprache-Befehle in den Speicher gehackt waren, verging eine Zeit und so manche Klippe mußte umschifft werden...

Wenn man heute daran zurückdenkt, weiß man woran es damals geklemmt hat: Kein Assembler, kein Maschinensprache-Monitor und keine Grundlagen-Artikel zur Hand. Einige quallvolle Umstiegsversuche folgten...

Im 64'er-Magazin stand der Rat: "Studieren Sie fremde Assembler-Listings...". Doch woher nehmen, wenn nicht stehlen! Im 64'er waren nur die MSE-Hex-Listings abgedruckt. Vielleicht per Befehlstabelle erst einmal "manuell disassemblieren"? Schnell stellte sich heraus – diese Methode ist auch nicht das Gelbe vom Ei! Mit viel Geduld und dem "Try+Error"-Prinzip gings dann langsam voran und die ersten Assembler-Projekte nahmen nach und nach doch noch Gestalt an.

Aus dieser Erfahrung und auf Grund zahlreicher Leseranfragen haben wir uns entschlossen, in diesem Monat dem Thema "Programmieren" (speziell in Assembler) einen großen Teil der Ausgabe zu widmen. Mit unserem Kurs "Von Basic zu Assembler" wollen wir allen Umsteigern und noch Unentschlossenen eine Brücke bauen, auf der Sie den sicheren Weg ins Lager der Assembler-Programmierer finden. Natürlich können Sie gleich loslegen, denn mit dem Maschinensprache-Monitor "SMON" haben Sie gleich das richtige Werkzeug zum Ausprobieren unserer Beispiele parat. Programmieren in Assembler ist gar nicht so kompliziert und hat viele Parallelen zum Fahrradfahren! Hat man die Sache mit dem Gleichgewicht im Griff, traut man sich immer mehr zu und irgendwann nimmt man auch mal die Hände vom Lenker! Sie werden sehen, die Programmierung in Assembler ist nicht schwer und kann höllischen Spaß machen!

Natürlich wollen wir die Anwender nicht im Stich lassen und haben auch diesen Monat tolle Software auf die Diskette zum Heft gepackt. Mit dem Funktions-Plotter "Geo-Funktion" und dem Mathe-Tool "Geometrie-Profi" werden Funktionen und geometrische Objekte anschaulich auf den Bildschirm gebracht. Genau die richtige Software für Mathematik-Fans und solche die es werden wollen! Spieler werden mit dem Tüftelspaß "Quadrant" auf Diskseite 2 verwöhnt. Hier gibt's ein packendes Spielprinzip, beeindruckende Grafik und umwerfenden Sound.

Viel Spaß beim Lesen, Programmieren, Rechnen und Spielen wünscht Ihnen

Ihr

Jörn-Erik B.

Jörn-Erik Burkert



GIG Süd e.V. – Bericht vom Frühjahrstreffen

Am 21.5.1995 fand das Frühjahrstreffen der GIG Süd e.V. in Buch am Buchrain statt, das unter den Mitgliedern und anderen Geos-Usern auf erfreulich hohe Resonanz stieß – während des Treffens konnte man vier neue Mitglieder werben. Besonders erfreulich: der Beitritt einer Geos-Userin (bislang gab´s nur ein einziges weibliches Vereinsmitglied). Mit von der Partie waren Rudolf Sanda und Christian Pichler von der GIG Wien sowie Rick Gaudet von CMD.

Ein Hauptthema war die Situation von Geos 3.0 (Falk Rehwagen will sich aus privaten und terminlichen Gründen aus der Programmentwicklung zurückziehen). Inzwischen ist bekannt, daß Wolfgang Grimm in seine Fußstapfen tritt und weitermachen wird.

Interessante Neuigkeiten hatten etliche Vereinsmitglieder in puncto Programmierung und Hardware-Basteleien mitgebracht:

- Erich Rupprecht fand heraus, daß ein Basic-Dreizeiler genügt, um Geos von der 1581 im Betrieb mit C 128D zu booten.
- Harald von Blumenthal ist gerade dabei, ein Verbindungskabel für den C 64 im Zusammenspiel mit beliebigen Modems zu entwickeln. Bei entsprechender Software arbeitete das Kabel bislang mit allen getesteten Modems.
- Auch eine spezielle Hardware-Konfiguration konnte man begutachten: C 64 mit Flash-8 und geoRam, angeschlossen an einer (auf 180 MByte aufgerüsteten) HD-40 und mit FDD-1541. Nach einigen Startschwierigkeiten lief das System hervorragend. Beispielsweise war der Bildaufbau unter GeoPublish (Text/Grafik) praktisch unmittelbar nach dem Anklicken abgeschlossen, Scrolling von Geo-Write-/GeoPaint-Dokumenten lief mit Super-Geschwindigkeit ab.

Schon während des Regio-Treffens der Gruppe Nürnberg konnte man diese Konfiguration bestaunen.

Wolfgang Petzold/bl

PC als "Sklave" des C 64

Bei Discount 2000 gibt´s ab sofort eine Kombination von Hard- und Software, mit der jeder C-64-User über eine 110 kBit/s schnelle serielle Datenverbindung Speicher und Ressourcen des PC nutzen kann: PC Slave, ein Steckmodul für den Expansionport. Ab sofort stehen Festplatten, Diskettenlaufwerke und Drucker- sowie V24-Schnittstellen des PC zur Verfügung.Zwei Server-Modi halten die Verbindung zum PC aufrecht:

1. Die Server-Software (PC Slave als TSR) läuft im Hintergrund – der PC kann zusätzlich andere Aufgaben erfüllen.
2. Als "dedicated Fileserver" liefert der PC bedeutend mehr Daten, läßt sich aber nur als Server einsetzen.

PC Slave braucht mindestens einen 80286er-PC und MS-DOS 5.0 (Test folgt). Inkl. 20seitigem Handbuch in Deutsch kostet das Modul 198 Mark.

Discount 2000, Am Wiesenpfad 1,53340 Meckenheim Tel. 02225/13360, Fax: 02225/10193

Pressekonferenz bei Escom

Fünf Wochen nach Commodores Versteigerung lud Escom zur ersten weltweiten Pressekonferenz nach Frankfurt ein:

- Commodore- und Amiga-Geschäfte werden streng getrennt: auf Intel-Prozessoren basierende PCs werden mit Commodore-Label vertrieben, der Amiga setzt weiterhin auf die bewährten Motorola-Zentraleinheiten. Nutzen will man alle sinnvollen Vetriebskanäle: Fachhändler, Kaufhausketten und Versandhandel.
- Neu gegründet: die Amiga Technologies GmbH, ein 100prozentiges Tochterunternehmen von Escom. Die Hardware-Entwicklung findet voraussichtlich in Philadelphia/USA statt, an neuen RISC-Amigas wird bereits intensiv gearbeitet und getüftelt.
- Der C 64 wird definitiv wieder hergestellt – an bewährten Produktionsstätten in China. Er soll vornehmlich in den asiatischen Markt kommen, auch im ehemaligen Ostblock (Rußland, Bulgarien, Rumänien, Polen) sieht man gute Absatzchancen für den legendären 8-Biter.

bl



Tecno Plus Control Pad

Neuheit von Data House: das Tecno Plus Control Pad TP511, speziell für C 64, Amiga und Atari. Es ist der Nachfolger des beliebten Swift-Pads TP200 (s. Testbericht in der 64´er 2/94). Im Vergleich zum Vorgänger besitzt es außer Auto- und Turbo-Feuer-Funktion ein bedeutend formschöneres und handlicheres Design, drei Feuer-Buttons und eine Schutzhülle für den Transport. Das Control-Pad kostet 19 Mark (ausführlicher Test in der nächsten Ausgabe).



Data House
Kai-Uwe Dittrich,
HarleshäuserStr.67
34130 Kassel,
Tel. 0561/68012,
Fax: 0561/68405

Brotkasten-Corner in Btx

Die "Brotkasten-Corner" (Btx-Befehl: *matting#), der C-64-Treffpunkt in Btx, wurde renoviert und aufgefrischt. Das Telesoftware-Angebot ist nun komplett überarbeitet.

Außerdem lassen sich jetzt auch Infos von "MegaCom-Software" abrufen – vor allem zu "Btx-Extra", dem "Highspeed"-Btx-Decoder für C 64/C 128. Programmierer Wolfgang Grimm (gleichzeitig auch Vertreiber) stellt inzwischen bereits die Beta-Version seiner Erweiterung für Geos 128 vor. *ma*

Neue Mailbox

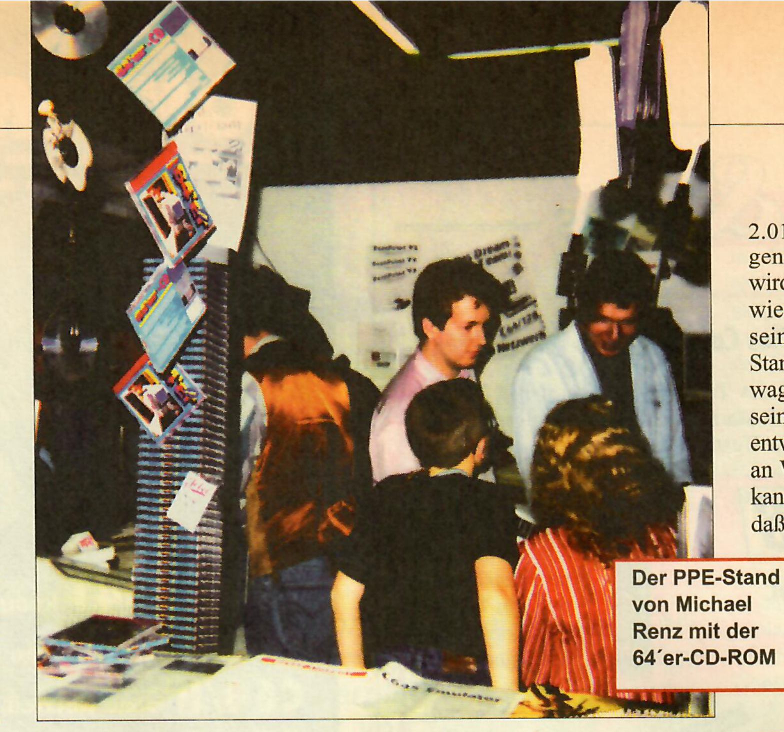
Ab sofort ist die "Omni World Germany" online. Die Mailbox läuft unter dem Programm "Omni-BBS 128" und ist über **Tel. 08121/79432** mit bis zu 14.400 bps erreichbar.Die Hardware-Ausstattung kann sich sehen lassen: C 128D mit 12,5 MByte RamLink und 85 MByte CMD-Festplatte. *ma*

CD-ROM Commander V1.0

Neu im Vertrieb von PPE M. Renz, Bornheim: der CD-ROM-Commander von Achim Täge.Die neuartige Software für den C 128 im 80-Zeichenmodus spricht SCSI-CD-ROM-Laufwerke an (auf dem Umweg über den Controller einer CMD-HD-Festplatte.Bei der Montage kommt man ohne Lötkolben aus. Anschließen lassen sich handelsübliche Single- oder Double-Speed-Laufwerke.Mit der Software des CD-ROM-Commanders lassen sich z.B. die Dateien der "64´er-CD-ROM" öffnen und auf ein beliebiges, Commodore-kompatibles Laufwerk (1541, 1571, 1581) speichern.

Der Preis für die Software inkl. Handbuch mit Montage-Anleitung steht noch nicht fest, soll sich aber zwischen 40 und 50 Mark einpendeln. Test folgt.

Performance Peripherals (Europe),
M. Renz, Holzweg 12, 53332 Bornheim



Der PPE-Stand von Michael Renz mit der 64'er-CD-ROM

Große Show für 8-Bit- und Bastel-Freaks

Hobbytronic '95

Für kommerzielle und private PC-Anwender ist die CeBIT Hanonover der Nabel der Computer-Welt. Homecomputer-Besitzer und Bastel-Freaks tummeln sich aber am liebsten auf der berühmten Elektronik-Messe in Dortmund. Möchten Sie uns bei unserem Messe-Bummel begleiten?

Anfang Mai dieses Jahres strömten Computer-Touristen massenweise in die Dortmunder Westphalen-Halle: die Hobbytronik öffnete ihre Pforten. Vor allem am Wochenende war's wie bei der CeBIT: total überfüllte Messehallen und kaum eine Chance, irgendwo einen Parkplatz zu bekommen.

Auf dem Weg zur Messe begegneten uns zahlreiche, schwer bepackte Computer-Freaks, die schon wieder auf dem Heimweg waren – es mußte sich also lohnen, einen ausgiebigen Blick auf die Messe-Neuheiten zu werfen! Es hat sich gelohnt - hier die Highlights, die wir auf unserem kurzen Streifzug entdeckten:

- Ein Team der "Enterprise"-Besatzung ließ sich anläßlich des 30jährigen Bestehens der "Star-Trek"-TV-Serie auf die Erde beamen und wurde staunend bewundert. Der "Enterprise"-Stand war der größte der Messe, das Star-Trek-Team hatte einen Flugsimulator mitgebracht und kredenzte den Erdenbewohnern den neuen Energiedrink "Warp4".
- Gleich nebenan wurde man ins Geheimnis von "Virtual Reallity" eingeweiht und persönlich in ein aktiongeladenes Spiel integriert (der Zuschauer sah das Game parallel dazu auf einem normalen Monitor). Allerdings - die Show warf uns nicht vom Hocker.
- In der nächsten Halle entdeckten wir eine informative Ausstellung zum Thema "Vorzeitliche Computertechnik - wie alles begann". Mancher moderne, ergonomisch gestylte Schreibtisch würde heute z.B. unter der Last eines fast 50 Kilo schweren Druckers enorm ächzen . . .



Hoher Besuch: A. Dettke, W. Kling und F. Rehwagen

Nahezu unüberschaubar war die Zahl der Händler, die auf der Hobbytronik die Gelegenheit nutzten, ihre Warenlager zu räumen. Da ließ sich manches Schnäppchen an Land ziehen: fast nagelneue C-64-Computer für 89 Mark, C-128-Tastaturen für fünf Mark und Mouse-Pads oder 10er-Packungen mit 5,25-Zoll-Disketten für zwei Mark!

Atari-Computer bot man teilweise schon für eine Mark pro Kilogramm an.

Es lohnte sich, die Preise zu vergleichen: so gab es gleichwertige Single-Speed-CD-ROM-Laufwerke schon ab 79 Mark, aber auch für 169 Mark.

Am interessantesten für C-64/C-128-Freaks waren zweifellos die Messestände des Geos-User-Clubs (GUC), Dorsten und von Performance Peripherals, Bornheim. Beide Aussteller waren oft derart umringt, daß man sich mit Mühe an die Verkaufsfläche vorkämpfen mußte – schließlich waren die beiden die einzigen, die ein aktuelles Angebot an C-64/C-128-Hard- und Software präsentierten. Der GUC stellte eine CD-ROM mit PD-/Shareware für Geoworks vor; die neueste Geoworks-Version 2.01 konnten wir allerdings nirgends entdecken – vermutlich wird Microsoft mit "Windows 95" wieder mal einen Tick schneller sein. Heiß diskutiert am GUC-Stand: Geos 64/128 3.0. Falk Rehwagen unterbricht vorübergehend seine Arbeit und gibt die Weiterentwicklung der aktuellen Fassung an Wolfgang Grimm weiter (man kann also durchaus damit rechnen, daß es doch noch eine verkaufsfähige Version des gepatchten Geos-Systems geben wird).

Am Samstag (13.5.95) traf hoher Besuch ein, der sich an den beiden genannten Messeständen zum Small-Talk und Fachsimplen traf: Arndt Dettke/Wolfgang Kling (GoDot), Wolfgang Grimm (Btx-Extra), Falk Rehwagen/Denis Döhler (GeoCom).

Messeneuheiten: "64NET" (um den PC am C 64/C 128 zu nutzen), ist jetzt endlich verkaufsfähig. Eine spezielle Geos-Anpassung und GoDot-Copys gehören dazu – die meisten PC-Grafiken lassen sich also unter GoDot direkt in den C 64/C 128 holen und dort weiterbearbeiten. Eine brandaktuelle GoDot-Version wird es bald bei PPE Renz mit ausführlichem Handbuch geben – Arndt Dettke denkt bereits über Version 2 (mit verbesserter grafischer Oberfläche) nach.

Wolfgang Grimm plant, sich des Geos 3.0 anzunehmen, will aber die Btx-Projekte uneingeschränkt fortsetzen: Btx für Geos und eine Anpassung an den neuen KIT-Standard.

Michael Renz (PPE) zeigte eine ans deutsche Geos-Kernel angeglichene Fassung von "GeoShell V2.2" (Befehlsoberfläche ohne Icons), die ab sofort zu haben ist.

Absoluter Verkaufsrenner war aber die "64'er-CD-ROM" (fast jeder Besucher wollte wissen, ob's noch weitere CD-ROMs geben wird). Ständig umlagert: der "64-Minitower" von PPE (s. Meldung in der 64'er 5/95). Eventuell wird es bald auch eine abgespeckte Version (ohne integrierte Floppy 1581) und eine aufgemotzte Fassung (mit Flash-8) geben.

Ein Gerücht hielt sich standhaft während der Messetage und beunruhigte die Commodore-Freaks: Escom will den C 64 doch nicht wieder herstellen, es ging bei der Übernahme von Commodore nur um den Namen. Nichts davon ist wahr: Anfang Juni erklärte Escom bei einer Pressekonferenz, daß der C 64 künftig in China für den asiatischen Markt und den ehemaligen Ostblock gebaut wird.

Denis Döhler/bl



Das Scanntronik-Team 1995

Jubiläumsfeier

Scanntroniks Party

Alles zu seiner Zeit: wer oft sieben Tage in der Woche unter Streß steht und hart arbeitet, darf auch mal feiern – vor allem, wenn's ums 10jährige Firmenjubiläum geht. Hubert und Renate Mugrauer luden zur Party – und alle kamen. Die 64'er -Redaktion war dabei . . .

Der 5.Mai 1995 war in unserem Redaktionskalender rot gekennzeichnet: um 16 Uhr sollte die Jubiläumsparty bei Scanntronik, Zorneding (vor den Toren Münchens) steigen.

Mitten in der Rush-Hour machte sich unsere Redaktion auf den Weg, begleitet von Albert Petryszin (der zuständige Mann für die Anzeigen im 64'er-Magazin) und Haus- und Hof-Fotograf Roland Müller (von dem stammen die Schnappschüsse auf dieser Seite). Obwohl's nur knapp sieben Kilometer Fahrtweg waren, kam unsere Crew zwanzig Minuten zu spät – dennoch waren wir die ersten (die anderen steckten offensichtlich in einem noch schlimmeren Stau als wir).

Endlich – gegen 17 Uhr waren alle Gäste eingetrudelt: das Fest konnte steigen. Die Gastgeber luden zu Sekt und Knabberleckereien (die von unserem Team fast bis zur letzten Erdnuß verputzt wurden, weil wie üblich das Mittagessen wieder ausgefallen war).

Insgeheim freute sich unser stellvertretender Chefredakteur Harald Beiler aufs Wiedersehen mit Hans Haberl, Scanntroniks Software-Produzent: beide bekamen 1985 erstmals Kontakt miteinander, als Beiler ihn um eine fertige Version des legendären Zeichenprogramms "Hi-Eddi" auf Diskette bat (und sich somit viele Stunden fehlerträchtiges Abtippen des in der 64'er 1/85 veröffentlichten Hexdump-Listings sparte) – schließlich wohnen beide im selben Ort.

Dipl.-Ing. Haberl präsentierte sich gut gelaunt und braungebrannt (ein Tribut an seine neue Wahlheimat Spanien), wobei sich im zwanglosen Gespräch schnell herausstellte, daß er trotz um sich greifender PC-Hysterie den C 64 noch lange nicht in die Ecke gestellt hat: nach wie vor verwendet er ihn vor allem in Verbindung mit seiner Amateurfunk-Anlage und zur Software-Entwicklung für Scanntronik-Projekte. Jüngstes Kind ist die C-64-Adaption eines komfortablen PC-Programms für Video-Schnittsteuerung (in Verbindung mit der PC-GenBox), das im Sommer 1995 fertig sein soll.

In seiner Festrede ließ Hubert Mugrauer nochmals rückblickend die Gründerjahre des Unternehmens Revue passieren und erzählte von der ersten Bekanntschaft mit Haberl: er sollte die Software zum legendären Super-Scanner entwerfen. Auch hier war "Hi-Eddi" der Anknüpfungspunkt für die ersten Kontakte.

Obwohl sich Mugrauer künftig verstärkt dem PC/AT-Markt widmen will (wobei sich Sohn Markus als fantastisches Programmiertalent entpuppt hat), bleibt der C 64 nach wie vor fester Bestandteil der Firmenpolitik: nachdem Escom bei der Übernahme von Commdores Erbe erklärt hatte, den C 64 künftig in China bauen zu lassen, um den asiatischen Markt damit zu erobern, wurde kurzerhand eine Reise nach China gebucht, um die Lage vor Ort zu sondieren (lesen Sie dazu unser Interview mit Hubert Mugrauer in der nächsten Ausgabe).

Ein fantastisches Vier-Gänge-Menü in einem renommierten italienischen Lokal beschloß die gelungene Jubiläumsfeier. *bl*



Die Damen der Belegschaft (ganz rechts: Renate Mugrauer) kümmern sich um den Bürokram



Trafen sich zuletzt bei der CeBIT'93 in Hannover: Hans Haberl und Harald Beiler



Hubert Mugrauer erläutert eine seiner ersten Hardware-Entwicklungen

Fühlt sich mit dem C 64 sichtlich wohl: Hans Haberl

Btx-Extra



Decoder der Extra-Klasse

Lang ist's her, seit der erste Btx-Decoder fÜr den C 64 auf den Markt kam. Dann kümmerte sich keiner mehr um das Produkt (lediglich die erste Version wurde ein wenig verbessert). Die neueste Fassung "Btx-Extra" übertrifft aber alle Erwartungen.



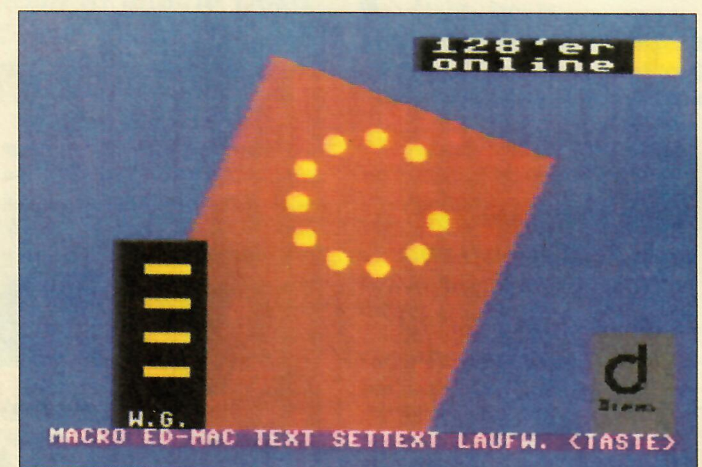
Zusätzliche Systemvoraussetzungen sind Floppy (1541,1571 oder 1581), RS232-Schnittstelle am Expansionport, Swiftlink oder Datablast, der BTX-Decoder (Version 1.6) von Drews und ein schnelles Modem mit mindestens 2400 bps. Wer an die Zukunft denkt, legt sich allerdings ein Modem mit bedeutend höherer Geschwindigkeit zu (z.B. mit 14.400 bps, ca. 200 DM. Natürlich kommen auch die User ohne Swiftlink nicht zu kurz.

Auf jeder Systemdiskette ist ein Startprogramm für die Userport-RS 232-Schnittstelle zu finden. Hier sind die Geschwindigkeiten aber eingeschränkt: der C 64 macht's mit 1200 bps, der C 128 mit 2400, wobei Telesoftware bei beiden nur mit 1200 bps funktioniert.Wie bereits erwähnt, lassen sich mit der normalen RS232-Schnittstelle am Userport nur Baudraten bis 1200 bps fehlerfrei fahren. Mit den RS232-Modulen am Expansionport, Swiftlink von CMD oder Datablast (PPE) sind aber höhere Geschwindigkeiten möglich. Bei der C-64-Version klappt es dann auch mit 2400 bps, außerdem ist der Bildschirmaufbau erheblich schneller. Die C-128- Version erreicht sogar 14.400 bps.

Der feine Unterschied

BTX-Extra bietet viele neue Funktionen, z.B. Programmstart von jedem möglichen Laufwerk (8 bis11), der Wechsel der Diskettenstation klappt jetzt auch im Online-Betrieb. Außerdem läßt sich das Inhaltsverzeichnis jedes aktivierten Laufwerks auf den Screen bringen und anzeigen.

Makros kann man nur Online erzeugen, da der Computer alle Eingaben mitschreibt. Sie dürfen 255 Zeichen lang sein und lassen sich auf Disk sichern. So spart man die zeitraubende Suche nach Btx-Nummern oder Anbietern.



Btx-Extra: die Erweiterung zum Drews-Btx-Decoder kann mit einigen Überraschungen aufwarten!

Ein kostensparende Lösung ist der integrierte Texteditor. Damit schreibt man Mitteilungstexte in aller Ruhe, speichert sie auf Disk und lädt sie später in die gewünschte Mitteilungsseite. Dabei stört ein wenig, daß der Editor nicht bei allen Btx-Seiten funktioniert (z.B INFO). Das liegt aber oft am skurrilen Seitenformat mancher Btx-Anbieter.

Telesoftware (TSW) läßt sich per Schnittstelle am Userport nur mit einem Speed von 1200 bps laden, mit "Swiftlink" schaffen C 64 und C 128 sogar 2400 bps.

Extra-Bonbon für C-128-Freaks

Speziell C-128-Usern bietet "Btx-Extra" viel: der Decoder arbeitet mit dem 80-Zeichenmodus – unterstützt werden sowohl der 16-KByte- als auch der 64-KByte-VDC-Chip. Positiv: wenn man mit verschiedenen Zeichensätzen arbeitet, stehen diese durch einfaches Umschalten per Tastendruck sofort zur Verfügung. Der zweite Btx-Sonderzeichensatz ist automatisch implementiert.

Die außergewöhnlich schnelle Grafikdruck-Routine bringt alle Btx-Seiten, Grafiken und Btx-Sonderzeichen in Sekundenschnelle zu Papier. Allerdings lassen sich keine VDC-Bildschirme als Grafik speichern, bei Textseiten funktioniert´s dagegen anstandslos. Das Programmpaket wird durch ein verbessertes Konfigurationsprogramm abgerundet.

Unseren Geschwindigkeitstest (s. Kasten) haben wir mit der Btx-Seite *343441182# durchgeführt. Es haben sich nahezu die gleichen Werte herauskristallisiert, die der Autor angibt.

Ein Tip für Swiftlink-User: die Baudraten lassen sich nicht ohne weiteres ändern, aber es gibt einen Trick: Laden Sie "BTX-START .. SL" und LISTen Sie das Programm. Ändern Sie jetzt die Werte hinter dem SYS-Befehl:

```
SYS .....,12 (C 64: 1200 bps)
SYS .....,24 (C 64: 2400 bps)
SYS .....,96,64 (C 128D: 9600)
SYS .....,14,64 (C 128D:14400)
```

Der zweite Parameter (64) repräsentiert die Kapazität des VDC-Chip (gegebenenfalls durch die Zahl "16" ersetzen!).

Geschwindigkeitstest

Version	Zeit
Btx 64 (1200/75)	47 sec
Btx 64 (2400)	26 sec
Btx 128 (2400)	21 sec

Auf einen Blick

Die Decoder-Erweiterung läßt nahezu keine Wünsche offen – dennoch sind uns doch ein paar negative Aspekte aufgefallen: der Texteditor ließe sich z.B. noch komfortabler gestalten. Je nach Anbieter ist nicht auf jeder Mitteilungsseite möglich, offline zusammengestellte Texte fehlerfrei zu laden. Zugegeben: die manchmal undefinierbaren Seitenformate der Btx-Anbieter sind oft nicht ganz schuldlos daran (z.B. bei *INFO#). Bei der Directory-Ausgabe ging´s uns ebenso: der Bildschirm wird nicht vollständig gelöscht (Reste der Verzeichnisliste blieben erhalten).

Die Abwahl (Verbindung trennen) klappt nicht immer: das Modem trennt die Seiten nicht korrekt, bei erneuter Anwahl kommt dann keine Verbindung zustande. Dieser Bug fiel uns bei der Abwahl mit der entsprechenden Funktionstaste F8 und bei "*9#" auf. Ungeklärt ist, ob´s wirklich an Btx-Extra liegt.

Die Anleitung auf Diskette wurde im Geos-Format gespeichert (was ist mit den Anwendern, die Geos 2.0/2.5 nicht besitzen?). Hier wäre simpler ASCII-Text bedeutend anwenderfreundlicher.

Peter Breuer/bl

64'er-Wertung: Btx-Extra 64/128

... ist die komfortable und gelungene Erweiterung des Btx-Decoders V1.6

Positiv

- leichte Installation
- benutzerfeundlich
- Editor für Textentwurf (offline)
- Laufwerkswechsel möglich
- Floppy-Befehle online
- gut verständliche Anleitung

Negativ

- Directory-Anzeige wird auf diversen Btx-Seiten komplett gelöscht
- Abwahl klappt nicht immer (Modem legt nicht auf)
- Anleitung als Geos-File

Wichtige Daten

Autor: Wolfgang Grimm
Bezugsquelle: Performance Peripherals Europe, M. Renz, Holzweg 12, 53332 Bornheim
Preis: 19,90 Mark (C-64-Version) 29,90 Mark (C 128-Version)
Testkonfiguration: C 128D (Blech), 1571, Monitor 1084s, Star LC24-100, Swiftlink bzw. REX RS232-Interface Faxmodem 14400. Software: Drews Btx-Decoder V1.6

Beurteilung:

SEHR GUT

Endlich da: die 64´er CD-ROM



120 64´er-Programm service-Disketten von 1984 bis 1994 als Disk-Images
100 64´er -Sonderheftdisketen
zwei C-64-Emulatoren für den PC: "Personal C 64" ,"C 64S"
ein Amiga-C-64-Emulator: "A64" (Shareware)
"64NET" (Demoversion): PC mit CD-ROM am C 64

nur 29,90 DM !

BESTELLCOUPON

Die 64er CD-ROM erhalten Sie bei (Vorkasse, Scheck oder Rechnung): zzgl. 6 Mark Versandkosten bzw. Nachnahmegebühr:



ERDEM Development, Postfach 1823, 84471 Waldkraiburg
Stonysoft, Beethovenstr. 1, 87727 Babenhausen

Ich bestelle:

------------- **64er CD ROM á 29,90 DM**

Name: ____________________

Straße, Hausnummer: ____________________

PLZ, Wohnort: ____________________

Telefon ____________________

Bitte ausschneiden und an eine der o.a.Adressen schicken!

Szene

Mit unserem Szene-Interview kommt ein wenig mehr Personality in unsere Rubrik. In dieser Ausgabe konnten wir "The Syndrom" ein wenig über ihn selbst ausquetschen. Zusätzlich haben wir die ersten Infos zur "Assembly 95" parat.

Stuff on Disk

Auch diesen Monat haben wir auf die Diskette zum Heft ein spezielles Bonbon der C-64-Szene gepackt - einen gelungenen Bitmap-Effekt aus dem neuesten Oxyron-Werk "Parts!". Einfach anschauen und staunen!
Wer an weiteren Demos und Diskmags interessiert ist, kann sich an folgende Adresse wenden:
Gonzo (AWT)
c/o Jörg Nehls
Marienenbergstr.12
31171 Nordstemmen
Bitte Leerdisketten und einen ausreichend frankierten Rückumschlag der Sendung beilegen. Ein kleines Geschenk (CDs o.ä.) wären als Dankeschön auch nicht schlecht.

Parts: Das neue Demo von Oxyron zeigt tolle Bitmap-Effekte auf dem Bildschirm

Die 64'er-Charts:

In der Szene kursieren zahlreiche Disk-Mags. Fast jede Gruppe hat ein eigenes Magazin auf Diskette. Wir haben uns aktuelle Mags angeschaut und deren Charts ausgewertet. Aus den einzelnen Wertungen haben wir die Over-All-Wertung ermittelt.
So geht's: Wir haben jeweils die fünf besten in den Kategorien **"Beste Demogruppe"**, **"Bester Coder"**, **"Bester Musiker"** und **"Bester Grafiker"** herausgezogen. Der erste Platz bekam fünf Punkte, der zweite vier, der dritte drei Punkte...
Die Rubrik **"Bester Cracker"** wurde von uns ganz bewußt ausgeklammert, da momentan das Thema Cracks und Raubkopien umstrittener denn je ist.
Außerdem gibt's auf dem C 64 ja wohl kaum noch Spiele zu knacken. Es sei denn einige Freaks "knacken" PD-Spiele oder aus dem 64'er-Magazin und bekleckern sich so mit "Ruhm".
Folgende Magazine wurden zur Ermittlung der Charts ausgewertet:
Relax #09
Chit Chat #7
Sky High #16
Nitro #18
Tribune #51
Vandalism News #20

Assembly 95

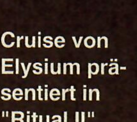
Die unbestreitbar bekannteste Computer-Party Europas findet jeden Sommer in Finnland statt. Auf der "Assembly" treffen sich C-64-, Amiga- und PC-Fans. Ein Wochende wird gecoded, gepixelt und komponiert, wobei unzählige Nebenveranstaltungen nicht nur zum Computern einladen.

Wie immer gibt es DEMO-Competitions, Grafik- und Soundwettbewerbe. Hier kurz die Regeln für die C-64-Competitions:

DEMO: Das Demo muß auf einem Standard-C-64 mit Floppy 1541 und einer Action-Replay-MK-6 laufen. Die maximal Dauer des Demos beträgt 15 Minuten.

GRAFIK: Alle Grafikdateien müssen sich ausführen lassen. Es werden alle Grafikformate akzeptiert, wobei Scoller oder ähnliche Elemente nicht erlaubt sind. Der Grafiker muß während der Party anwesend sein.

Sound: Das Musikstück darf nicht länger als drei Minuten dau-



Cruise von Elysium präsentiert in "Ritual II" seine schönsten Bilder





Beste Demogruppe

Platz	Name	Punkte
1 (1)	Oxyron	29
2 (2)	Camelot	17
3 (-)	Byterapers	11
4 (5)	Fairlight	9
5 (-)	Refelx	6

Bester Coder

Platz	Name	Gruppe	Punkte
1 (1)	Slammer	Camelot	21
2 (2)	TTS	Oxyron	29
3 (-)	Mr.Sex	Byterapers	16
4 (3)	Crossbow	Crest	10
5 (4)	MMS	Taboo	6

Bestes Disk-Mag

Platz	Name	Gruppe	Punkte
1 (1)	Skyhigh	Oxyron	24
2 (2)	Nitro	Excess	19
3 (3)	Shout!	Equinoxe/Fairlight	9
3 (4)	Reformation	Fairlight	9
5 (-)	Revealed	Oxyron	8

Inside

ern, wobei Samples erlaubt sind. Der Künstler muß auf der Party anwesend sein.

Die Organisatoren der C-64-Competition in diesem Jahr sind: Deadbeat/ The Sharks, Mysdee/ The Sharks und Hazor/ Beyond Force. Den Gewinnern winken folgende Geldpreise:

DEMO: 1. 200$, 2. 100$, 3. 50$
MUSIK: 1. 100$, 2. 50$, 3. 25$
GRAFIK: 1. 100$, 2. 50$, 3. 25$

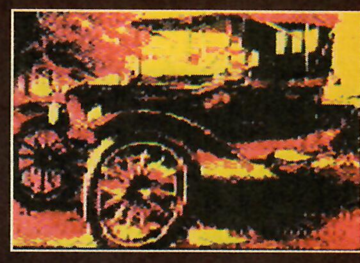
Neben den üblichen Competitions haben die Veranstalter außerdem noch eine Video- und Musikanlage organsiert. Für heiße Abende sorgen Rockkonzerte und Techno-Raves in benachbarten Hallen. Außerdem findet Finnlands größte Home-Computer-Messe in direkter Nachbarschaft zur Assembly statt.

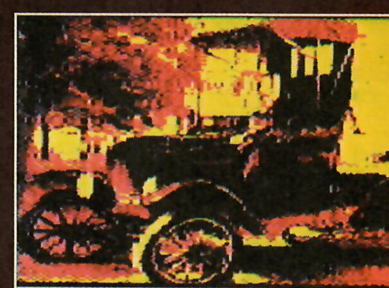
Das ganze Spektakel läuft vom *10 bis 13 August 1995 im Helsinki Fair Centre.*

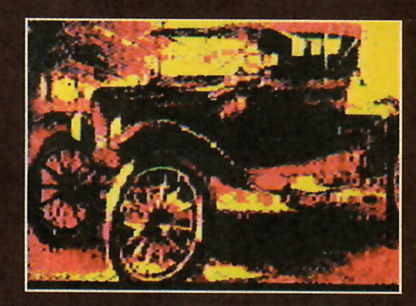
Der Eintritt kostet ca. 55 US-Dollar. Für nähere Informationen stehen die Organisatoren gern zur Verfügung:

Assembly 95
Lakkisepant 12
00620 Helsinki
FINLAND

Telefon: ++358 (0) 777 3721
++358 (0) 757 3115
Mailbox: ++358 (0) 615 000028
(STARPORT, 4 Leitungen PC)
++358 (0) 615 000029
(STARPORT, 5 Leitungen PC)
Internet: telnet -8 mpoli.fi -l pcboard
tp.mpoli.fi/starport/asm95
http:/www.icon.fi/assembly95
kemu@zombie.oulu.fi
kemu@nic.funet.fi



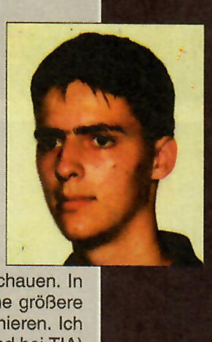




Auch ein Oldtimer wird in Parts von Oxyron "verbeult"...



Die 64'er-Szene hat viele Stars und Sternchen. Wer sich hinter den Szene-Namen aber versteckt, wissen die wenigsten. Wir wollen einige bekannte Szener ab dieser Ausgabe vorstellen. Für dieses Heft hatten wir "The Syndrom" an der Strippe. Er ist schon seit einigen Monaten in den 64'er-Szene-Charts die Nr. 1 bei den Musikern.



64'er: Stelle Dich kurz vor!
S: Mein richtiger Name ist Matthias Hartung. Ich bin 21 Jahre alt und studiere Wirtschaftsinformatik im 4.Semester in Dresden.
64'er: Wie bist Du in die Szene gekommen?
S: Im Herbst 1988 habe ich mit ein paar Freunden angefangen, mehr oder weniger tolle Demos zu machen, die erst unter ECC und später unter TIA releast wurden. Nach der Grenzöffnung ergab sich der Kontakt zu anderen Gleichgesinnten fast von selbst. Da wir später keinen Musiker finden konnten, habe ich selbst angefangen, dem C 64 Töne zu entlocken. Mittlerweile ist TIA ziemlich bekannt.
64'er: Wie arbeitest Du und welche Programme nutzt Du?
S: Die meisten meiner Musiken mache ich "am Stück", d.h. wenn ich eine Idee habe, wird diese umgesetzt, ohne dabei auf die Uhr zu schauen. In letzter Zeit war ich aber mehr auf dem Spielesektor aktiv, da es eine größere Herausforderung ist, den Soundtrack passend zum Spiel zu komponieren. Ich benutze hauptsächlich die TIA-Musik-Editoren, die von BRIAN (Mitglied bei TIA) und mir programmiert wurden. Zur Zeit arbeite ich gerade an einem neuem Musikplayer.
64'er: Warum "The Syndrom"?
S: Das hat eigentlich nichts besonderes zu bedeuten. Als wir anfingen, Demos zu machen, brauchte natürlich jeder ein Pseudonym, und mir kam irgendwie der Name des Spiels "Alien Syndrome" in den Sinn und so wurde dann "The Syndrom" daraus.
64'er: Jetzt einige Fragen für Deine Geschmacksnerven:
Getränk: Sprite
Essen: Spaghetti, Pizza
Farbe: blau
Musik: Acid, Hardtrance
Film: alle ZAZ-Produktionen
Auto: VW Golf III
Frau: Meg Ryan
Computer: C 64
Buch: alle Karl May-Bücher
Comic: Donald Duck
64'er: Was wünschst Du Dir für den C 64 in Zukunft?
S: Es wäre erfreulich, wenn der Markt wieder etwas in Schwung käme, weil dadurch auch wieder gute Spiele produziert würden. Außerdem hoffe ich, daß die 64'er noch lange durchhält, denn ohne das 64'er-Magazin wäre der C 64 zeitungsmässig wohl ziemlich alleingelassen.
64'er: Vielen Dank für das Gespräch!



Fast immer unter den besten fünf Grafikern im 64'er-Magazin: Cruise von Elysium

Bester Grafiker

Platz	Name	Gruppe	Punkte
1 (1)	Electric	Extend	27
2 (2)	Creeper	Antic	26
3 (3)	Ogami	Fairlight	14
3 (4)	Cruise	Elysium	12
3 (5)	Joe	Wrath Design	7

Bester Musiker

Platz	Name	Gruppe	Punkte
1 (1)	The Syndrom	Crest/TIA	27
2 (2)	PRI	TIA/Oxyron	26
3 (3)	Jeff	Camelot	16
4 (4)	Drax	Crest	6
5 (5)	Jereon Tel	Maniacs of Noise	5

AKTION!

Software-Klassiker auf Diskette

64'er-Assembler-Paket

Das ist unser Service für alle Leser, deren 64'er-Software-Sammlung noch Lücken hat: das Assembler-Paket ist eine Sammlung aus Assemblern, Monitoren und vielen Hilfsprogrammen auf einer Diskette.

Assembler auf Disk

Wer einen Assembler sucht, kann gleich zwischen drei unterschiedlichen Programmen wählen:

1. Hypra-Ass
2. Giga-Ass
3. VIS-Ass V.6

Während "Hypra-Ass" und "Giga-Ass" den Basic-Interpreter für die Eingabe nutzen, hat "VIS-Ass" einen eigenen Editor mit umfangreichen Befehlssatz. Alle drei Assembler verarbeiten Macros und können bedingt Assemblieren. Der "VIS-Ass" hat neben seinem umfangreichen Editor eine Oberfläche integriert, die mit einem Mauspfeil per Joystick bzw. Maus bedient wird.

Die Maschinensprache-Monitore

Auf der Diskette finden Sie den legendären SMON in verschiedenen Ausführungen:

1. SMON classic
2. SMON II
3. SMON + (mit Disk-Monitor)
4. SMON 128D



Mit Final Mon haben Sie den C-64-Speicher voll im Griff

Die Assembler auf der Diskette (hier Giga-Ass) ermöglichen komfortables Programmieren

Zusätzlich zu den vier SMON-Versionen finden Sie noch den Maschinensprache-Monitor "Final Mon", der noch einige Befehl zusätzlich parat hat.

Tools, Konverter und noch mehr

Um die Arbeit mit den Maschinensprache-Tools zu vereinfachen, finden Sie einige Hilfsprogramme auf der Diskette:

Reassembler: Der "VIS-REASSEMBLER" wandelt lauffähige Maschinensprache-Programme in VIS-Ass-kompatiblen Quellcode um. Die gleiche Aufgabe übernimmt der "REASS". Er reassembliert Object-Files in Hypra-Ass-Quelltexte, die Sie ohne Probleme mit dem Hypra-Assembler editieren und wieder zu lauffähigen Code assemblieren können.

Source-Konverter: Sie möchten einen Giga-Ass-Quelltext mit VIS-Ass weiterbearbeiten? Kein Problem - das Programm "GIGA-VIS-CONVERTER" wandelt Source-Codes vom Giga- ins VIS-Format. Wenn Sie mit Hypra-Ass-Quelltexte arbeiten möchten, nutzen Sie zuvor den "HYPRA-GIGA-CONV". Er übersetzt die Hypra-Ass-Quelltexte automatisch ins Giga-Ass-Format.

Assembler-Editor: Mit dem Editor zum Hypra-Ass bekommen Sie ein hilfreiches Werkzeug zum editieren Ihrer Assembler-Listings. Die Funktionstasten sind hier mit Befehlen belegt und der Quelltext läßt sich beliebig in vertikaler Richtung gescrollen.Für die Bestellung der Extra-Diskette verwenden Sie bitte den Coupon. Es genügt selbstverständlich auch eine formlose Benachrichtigung (Brief oder Postkarte), wenn Sie das Heft nicht zerschneiden möchten. Viel Spaß mit unserem Assembler-Paket! *lb*

READ.ME-Datei mit Druckausgabe

Eine umfassende Anleitung zu diesem Software-Produkt finden Sie auf der Diskette.
Dazu lädt und startet man:

```
LOAD "READER V1",8
```

und startet mit RUN.
Die Optionen des Hauptmenüs (zu den einzelnen Menüpunkten kommt man mit den Cursor-Tasten aufwärts/abwärts):

Floppy: Nach dem Tipp auf <RETURN> bringt der Screen das Directory. Interessant sind hier lediglich die Dateien mit der Endung ".TXT". Bewegen Sie den Auswahlbalken per <CRSR auf/ab> und laden Sie den gewünschten Anleitungstext mit <RETURN>.

Text: *Lesen:* ... bringt die erste Bildschirmseite, geblättert wird ebenfalls mit den Cursor-Tasten auf/ab. Mit <RUN/STOP> bricht man ab und kehrt ins Hauptmenü zurück.
Bei *Suchen:* Geben Sie einen gewünschten Begriff ein (z.B. einige Buchstaben, ein Wort oder einen ganzen Satz). Nach kurzer Zeit meldet sich der Computer wieder mit der ersten Bildschirmseite, der Suchbegriff ist jetzt aber im folgenden Gesamttext weiß markiert.

Printer: ... schickt den Text in 40-Spaltenbreite zum Drucker. Vorher stellt man im Druckermenü ein, ob's ein seriell angeschlossenes Commodore- bzw. Epson-kompatibles Gerät ist, oder ob man statt dessen mit einem Parallelkabel am Userport (verbunden mit der Centronics-Schnittstelle) arbeitet. Gegegebenenfalls legt man fest, ob ein Zeilenvorschub (Line Feed, LF) gemacht werden soll.

Programmende: Damit kehren Sie wieder in den Direktmodus des Computers zurück. Die auftauchende Fehlermeldung "Syntax Error" ist bedeutungslos.

BESTELLCOUPON

☐ Ja, ich bestelle die Software-Klassikerdisk mit Anleitung:
64'er 7/95: 64'er Assembler-Paket

___ Stück 5,25-Zoll-Diskette (beidseitig bespielt) zum Preis von 9,80 Mark

☐ Ich bezahle den Betrag zzgl. 6 Mark Versandkosten

☐ nach Erhalt der Rechnung ☐ per Scheck anbei

Name: ______________________

Straße, Hausnummer: ______________________

PLZ, Wohnort: ______________________

Datum/Unterschrift: ______________________

Schneiden Sie bitte den ausgefüllten Bestellcoupon aus, kleben Sie ihn auf eine Postkarte und schicken Sie ihn an:
64'er-Magazin Leserservice, D-74170 **Neckarsulm**,
Telefon: 0 71 32/9 69-185
oder bequem per Fax: 0 71 32/9 69-190

Durchblick garantiert!



NEU

Sonderheft Grafikkarten & Monitore

Ab 21. Juni '95 überall im Handel



Folge 1

Von Basic zu Assembler

Programme auf dem C 64 sind in Basic blitzschnell geschrieben und lauffähig. Wenn der C 64 aber Ermüdungserscheinungen zeigt und in Basic schlapp macht, muß Maschinensprache her. Unser Kurs wird Ihnen beim Umstieg in das C-64-Turbo-Zeitalter helfen.

Für fast jeden BASIC-Programmierer wirkt Maschinensprache auf den ersten Blick sehr kompliziert, da sich jeder BASIC-Befehl teilweise aus mehreren Assembler-Anweisungen zusammensetzt. Wenn die Angelegenheit so umständlich ist, warum sollte man dann aber umsteigen? Maschinensprache ist im Gegensatz zu Basic viel schneller und spart Speicher (s. Kasten "Maschinensprache - schnell und sparsam mit Speicher"). Viele Berechnungen und Effekte lassen sich nur durch direkte Programmierung in Assembler realisieren.Bevor wir aber zur Tat schreiten und uns in die Tiefen der Maschinensprache begeben, laden Sie bitte den SMON von der Diskette zum Heft oder installieren Sie einen Maschinensprache-Monitor Ihrer Wahl.

Jetzt geht's los!

Den Umstieg von Basic zu Assembler starten wir an Hand des POKE-Befehls. Diese Anweisung macht bekanntlich nichts anderes, als eine 8-Bit-Zahl einer Speicherstelle zuzuweisen. Mit POKE 53280,0 wird der Bildschirmrahmen schwarz gefärbt. Wie funktioniert das in Assembler? Dazu aktivieren Sie im SMON den Assembler-Mode mit der Anweisung *A 1000*. Nun können wir Befehle eingeben, die der SMON in Prozessor-Codes übersetzt.

Tippen Sie folgende Anweisungen ein und bestätigen Sie jede Zeile mit der RETURN-Taste:

LDA #$00
STA $d020
RTS

Mit *G 1000* können Sie nun die kleine Routine starten und feststellen, daß der Bildschirmrahmen schwarz ist. Um das Programm zu verstehen, müssen Sie wissen, daß der Prozessor des C 64 mit drei Registern arbeitet, die alle acht Bit groß sind.

Beim Übertragen des POKE-Befehls haben wir uns dem A-Register bedient. Es wird in der Fachsprache Akkumlator (Akku) genannt - es ist das Hauptregister. Der Befehl

LDA #$00

beschreibt den Akku mit dem Wert Null und bedeutet "LADE AKKU MIT". Um den POKE-Befehl zu komplettieren, muß nun der Wert 0 in die Speicherstelle für die Bildschirmrahmenfarbe geschrieben werden. Dazu dient der zweite Befehl in unserem kleinen Mini-Programm. Durch

STA $d020

wird der Wert im Akku in die Speicherzelle 53280 (hex. $d020) übertragen und die Farbe des Bildschirmrahmens auf Schwarz gesetzt. Der STA-Befehl schreibt also Werte aus dem Akku in Speicherstellen. Der letzte Befehl in der Liste sorgt für die Rückkehr aus dem Programm und ist mit dem BASIC-Befehl "RETURN" vergleichbar. Er sollte immer am Ende einer Assembler-Routine stehen, um eine sichere Rückkkehr ins Basic zu gewährleisten. Wer den Wunsch nach einer direkten Rückkehr zum SMON hat, sollte für *RTS* die Anweisung *BRK* eingeben. Sie sendet ein Break-Kommando und sorgt für den Rücksprung zum Maschinensprache-Monitor.

Um die Funktionstüchtigkeit unseres Programms außerhalb des Monitors zu testen, verlassen wir ihn mit dem X-Befehl und setzen den C 64 mit der Tastenkombination RUN/STOP+RESTORE in den Ausgangszustand. Nun starten wird unser kleines Maschinenprogramm mit:

SYS 4096

Der Bildschirm färbt sich schwarz und das Programm kehrt zur READY-Meldung zurück. Unser erstes kleines Assembler-Programm steht!

Als Übung versuchen Sie das kleine Programm so zu ändern, daß die Farbe des Bildschirm-

Hintergrunds auch schwarz wird. Dazu ändern Sie im STA-Befehl nur die Adresse (s. POKE) und starten das Programm erneut - schon wird auch der Background schwarz. Wenn Sie nicht auf Schwarz stehen, können Sie gern ein wenig mit dem LDA-Befehl experimentieren und eine neue Farbe aktivieren.

Weitere Register

Wie erwähnt, besitzt der Prozessor des C 64 drei Register. Den Akku kennen wir schon, es fehlen also nur noch seine beiden Kollegen. Sie werden mit X und Y bezeichnet. Ihre Verwendung hat viele Parallelen zum Akku, wobei es trotzdem einige Ausnahmen gibt. Sie lassen sich ebenfalls mit einem Wert beschreiben und der Inhalt in eine Speicherstelle übergeben. Aus dieser Tatsache ergeben sich vier neue Befehle:

LDX	*Lade X-Register*
LDY	*Lade Y-Register*
STX	*Schreibe X-Register*
STY	*Schreibe Y-Register*

Um ihre Wirkungsweise zu testen können Sie mit Hilfe des SMON unser kleines Miniprogramm so verändern, daß die Operationen nun mit dem X- oder Y-Register ausgeführt werden.

Weiter mit Basic

Nachdem wir die beiden neuen Register des C-64-Prozessors kennengelernt haben, wollen wir uns einem weiteren Basic-Befehl zuwenden. Als nächstes wird *PEEK* in Maschinensprache übertragen. Der Befehl liest bekanntlich den Inhalt von Speicherstellen. Er läßt sich durch eine einzelne Assembler-Zeile ersetzen. Wir wollen als Beispiel den Wert aus 53280 (Rahmenfarbe) ermitteln:

LDA $d020
RTS

Der kleine "Programmfetzen" holt sich aus $d020 (dez. 53280) den Wert für die Rahmenfarbe in den Akku und kehrt zurück. Das Beispiel zeigt eine neue Form des LDA-Befehls. Bei unserer kleinen POKE-Simulation wurde durch die Raute (#) ein direkter Wert in den Akku übertragen, jetzt wird der Inhalt der Speicherzelle $d020 ausgelesen und im Akku gesichert. Wir weisen also keinen Wert zu, sondern lesen den Inahlt einer Speicherzelle! Man sagt auch, daß der LDA-Befehl unterschiedlich adressiert wurde. Dazu aber später mehr, denn die Problematik würde im Moment zu sehr verwirren und wird deshalb zurückgestellt.

Läßt sich die neue Adressierungsart vielleicht auch auf die beiden anderen Prozessor-Register übertragen? Um diese Überlegung zu überprüfen, starten wir den SMON und verändern unser Programm:

LDX $d020
STX $d021
RTS

Mit *G 1000* starten wir das kleine Programm und Sie werden sehen, daß sich die Farbe des Bildschirm-Hintergrunds dem Rahmen anpaßt. Im Beispiel wurde mit dem LDX-Befehl der Wert aus 53280 (hex. $d020) gelesen und mit der folgenden Anweisung ins Register für die Hintergrundfarbe geschrieben. Unser zweites kleines Programm ist komplett und als Übung wollen wir nun endlich ein Wort auf den C 64 Bildschirm schreiben.

Dazu brauchen wir unser kleines Programm nur ein wenig zu modifizieren. Im ersten Abschnitt wurde festgestellt, daß der POKE-Befehl Speicherzellen mit Werten beschreibt. Der Bildschirmspeicher ist eine Ansammlung von Speicherzellen, die sich beschreiben lassen. Der Beginn des Bildschirm liegt bei 1024 (hex. $0400) und um ein "A" in die linke obere Ecke des Bildschirms zu schreiben bedarf es nur dem Basic-Befehl:

POKE 1024,1

Es dürfte nun für Sie kinderleicht sein den Bildschirm mit Texten Ihrer Wahl zu "pflastern". Probieren Sie doch ein wenig...

Ein Index kommt ins Spiel

Sicher ist es Ihnen nicht entgangen, daß man bei langen Texten viele LDA- und STA-Anweisungen eintippen muß. Das nervt auf die Dauer und verschlingt enorm Speicherplatz. In Basic würde man eine FOR-NEXT-Schleife benutzen. Listing 2 zeigt, wie in Basic ein Balken in der ersten Bildschirmzeile erzeugt wird. Listing 1 die gleiche Operation in Assembler.

Sie sehen sicher auf den ersten Blick, daß einige neue Befehle dazu gekommen sind. Als erstes wollen wir uns mit:

STA $0400,X

auseinandersetzen. Hier wird der Wert aus dem Akku in Adresse $0400 (dez. 1024) geschrieben. Zur festen Adresse im Argument ($0400) addiert der Prozessor aber noch den Wert aus dem X-Register. Beim ersten Druchlauf also $27 (dez. 39). Damit haben wir eine weitere Adressierungsart des STA-Befehls. Sie wird als direkt X-indiziert bezeichnet. Der aufmerksame Leser wird sich jetzt fragen, ob die ganze Prozedur auch mit dem Y-Register funktioniert! Studieren Sie den folgenden Abschnitt und probieren Sie es dann selbst!

Nun aber erst einmal die beiden neuen Befehle in unserem kleinen Beispiel. Der Befehl *DEX* subtrahiert den Wert 1 vom X-Register und steht für "Dekrementiere X-Register". Es entspricht im Prinzip der Laufvariable X in Listing 2.

Die folgende Anweisung ist ein sogenannter Branch- oder Verzweigungsbefehl. Er testet auf das letzte Ereignis. In unserem Falle bedeutet *BPL:* verzweige zu $1004 so lang das Ergebnis der Operation DEX positiv ist. Damit schließen wir den Kreis und unsere kleine FOR-NEXT-Schleife in Maschinensprache ist komplett.

Als nächstes ersetzen Sie die Befehle, die sich auf das X-Register beziehen durch Anweisungen mit Y und probieren die Funktionsfähigkeit. Sie werden sehen, daß die vorgestellten Befehle sich auch mit dem Y-Register realisieren lassen.

```
10 FOR X=0TO39
20 POKE1104+X,64
30 NEXT

.M 0800 0830
:0800 00 0E 08 0A 00 81 20 58   .....  X
:0808 B2 30 A4 33 39 00 1D 08   .0.39...
:0810 14 00 97 31 31 30 34 AA   ...1104.
:0818 58 2C 36 34 00 23 08 1E   X,64.#..
:0820 00 82 00 00 00 58 00 86   .....X..
:0828 20 00 00 00 39 76 00 40    ...9..@
.
.D 1000 1010
,1000  A2 27      LDX #27
,1002  A9 40      LDA #40
,1004  9D 00 04   STA 0400,X
,1007  CA         DEX
,1008  10 FA      BPL 1004
,100A  60         RTS
----------------------------------------
```

Basic-Programm (rot) und Assembler im Monitor unter der Lupe: Basic benötigt genau 25 Bytes mehr für die gleiche Operation

Maschinensprache - schnell und sparsam mit Speicher

Die auf den ersten Blick ein wenig komplizierten Assembler-Befehle versauern so manchen ambitionierten Basic-Programmierer den Umstieg auf die Maschinenebene. An Hand von Listing 1 und 2, wollen wir die Vorteile von Assembler zeigen. Dazu laden Sie den SMON und starten ihn mit SYS 49152. Nun geben Sie Listing 1 ein. Danach verlassen Sie den Monitor und geben das Basic-Programm (Listing 2) ein. In Zeile 30 ändern Sie die Adresse von 1024 auf 1104. Zusätzlich fügen Sie die Zeile 50 an:
50 SYS 4096
Nun können Sie das Basic-Programm mit dem RUN-Befehl starten. In der dritten Bildschirmzeile wird nun per FOR-NEXT-Schleife der waagerechte Balken auf dem Bildschirm aufgebaut. Sofort danach per Assembler. An dieser Stelle sorgt der Praxistest anschaulich, welchen Geschwindigkeitszuwachs in Assembler möglich ist.
Die Speicherplatz-Einsparung in Assembler ist enorm. Wenn Sie mit Hilfe des SMON (s. Bild oben) die Länge der beiden Programm vergleichen, werden Sie schnell erkennen, daß das Basic-Programm von $0801 bis $0825 belegt. Das Maschinenprogramm inklusive Rücksprungbefehl nur den Bereich von $1000 bis $100C beansprucht. Das sind bei einem kleinen Programm schon 25 Byte.

Noch mehr Text

Mit dem Strich am Bildschirm werden Sie sicher nicht so recht zufrieden sein und wollen Text Ihrer Wahl auf den Bildschirm bringen. Dazu erweitern wir unser kleines Programm. Um unterschiedliche Zeichen in der Variable A (Listing 2) im POKE-Befehl zu realisieren, arbeitet man mit dem READ-Befehl. Listing 3 zeigt, wie man das Wort "Test" auf den Bildschirm bringt. In Listing 4 sehen Sie die Lösung in Assembler. Die neue Form des LDA-Befehls sorgt für das Lesen der Zeichen ab $100c. Er arbeitet anlog zur STA-Anweisung und kann sowohl mit X- und Y-Register benutzt werden.

Farbe kommt ins Spiel

Natürlich kann es möglich sein, daß die beschriebene Textausgabe bei älteren C-64-Modellen nicht funktioniert. Dazu muß man bemerken, daß bei den Modellen, die schon einige Jahre mehr auf dem Buckel haben, das Farb-RAM extra gesetzt werden muß. Um Programme zu allen C-64-Modellen kompatibel zu machen, sollten Sie immer eine Routine zum Setzen des Farbspeichers in Ihre Programme integrieren. Der Bereich für den Farbspeicher be-


Wow! Soviel
Top Themen p
64'er
Sonderhefte
ab sofort
pro Heft nur DM 10,- (statt DM 16,-) nur solange Vorrat reicht!
Einsteiger
SH 50: Starthilfe
Alles für den leichten Einstieg/ Heiße Rythmen mit dem C64/ Fantastisches Malprogramm
SH 62: Erste Schritte
Exbasic Level II: Die Super-Basic-Erweiterung RAM-Exos: Floppy-Speeder der Extraklasse
SH 74: Einsteiger
Basic 3.5.: Basic-Erweiterung mit Hires-Grafik-Befehlen / FOBS V1.0: Floppy Operationen per Benutzeroberfläche
Anwendungen
SH 68:
Faszination Sternenhimmel. Der Blick in den Kosmos/ Kreuzworträtsel zum Selbermachen
SH 78:
Grafic-Calc:Grafische Auswertung von Jahresbilanzen/MAS V1.0: Übersichtliche Schulnotenverwaltung
SH 81:
Paint Mania: Zeichenprogramm der Superlative/ Maestro: Eigenen Sound auf Knopfdruck komponieren/ Disk Tools V6.5: Enttarnt jedes Byte auf Diskette
SH 86:
Database 2.0: Universelle Datenbank mit starken Rechenfunktionen/ Stamp Collection Kit: Archiv für Ihre Briefmarken/ außerdem: Autokauf, Haushaltskasse & Girokonto
SH 91:
RB-Fahrrad: Fahrtenbuch und Kostenverwaltung für Biker/ Video-Manager 3.0:erfaßt und verwaltet bis zu 600 Filme/ Sir-DOS V1.2: PC-Simulator für den C 64
GEOS
SH 48: GEOS-Erweiterungen
Geotext - neuer, schneller Texteditor für Geowrite/Workshop zu Geopublish
SH 59: GEOS
GeoBasic: Großer Programmierkurs mit vielen Tips & Tricks
SH 80: GEOS
Lottoblock: Statistische Gewinnzahlauswertung mit Tippvorschlägen/ Finanzen: Welche Geldanlage ist die Beste ?
SH 92: GEOS
Geo-Mensch: alles über den menschlichen Körper in Super-Grafik/ 14 optimale Druckertreiber für 24-Nadler/ 52.Master Copy: vollständiges Backup einer Geos-Disk in 52 Sekunden
SH 96: GEOS
Evolution: fantastische Reise in die Zeit der Dinosaurier und Mammuts/ Disketten-Dok: Reparatur und Sicherheitskopien beschädigter System-Disketten/Mega-Scrap: Grafikaustausch zwischen Geos- und Hi-Eddi-Format
Tips, Tricks & Tools
SH 65:
Streifzug durch die Zeropage/ Drucker-Basic: 58 neue Befehle zur Printer-Steuerung/ Multicolorgrafiken konvertieren/ über 60 heiße Tips&Tricks
SH 77:
Amica-Konvert: 6 Malprogramme tauschen Grafik aus/ Disc-Basic: Floppybefehle kurz und prägnant prüfen
SH 93:
Disk-Racoon 3.21: Diskettenverwaltung mit mausgesteuerter Benutzeroberfläche/ Basic-Expansion: 42 neue Befehle für Grafik und Floppy/ Diashow-Maker: perfektes Multimedia-Feeling
Hardware
SH 67:
Wetterstation.Temperatur, Luftfeuchtigkeit und -druckmessen/ DCF Funkuhr und Echtzeituhr/ Daten konvertieren: vom C64 zum Amiga, Atari ST und PC
SH 83: Floppy
4 Kopiertools für komplette Backups und Einzel-Files/ Spur 36 bis 40: 95 Blocks zusätzlich auf Diskette viele Tips&Tricks
SH 84: Hardware
C64 kompakt: Computer und Floppy in einem Gehäuse (Umbauanleitung)/ Midi-Interface: C64 mit Keyboard/ Tiny-EPROMer: EPROM-Brenner im Selbstbau unter 30 DM
Alle Hefte mit randvoller Programm-Diskette
DISKETTE IM HEFT
64'er Spiele
11 Superspiele für Drahtseilnerven
Daten-Agent 00X: Jagd im All
Zoff im Geisterschloß
Charakter-Editoren: Spielelösungen, Pokes und Cheats
Alle Programme auf Diskette
64'er HARDWARE
Floppy-Speeder für 10 Mark
8-Kanal-Schaltinterface
Sieben Sinne für den C 64
Der Edel-C 64 mit Floppy
10 Bauprojekte mit Programmen
Top Spiele mit Diskette
ACTION
JUMP'N RUN
LOGIK
TRICKS

Know how! plus Diskette!

Programmier-Sprachen

SH 71: Assembler
Kursus/ Komplettpaket/ Befehlsposter/ Tips & Tricks/ Leserfragen

Grafik

SH 55:
Amica Paint: Fantastisches Malprogramm für Hobby-Grafiker, mit allen Up-Dates

SH 75:
Interlace 64: 136 Farben und 640x200-Pixelgrafik und 80-Zeichen-Bildschirm/ 3D-Animation mit Hi-Eddi

SH 87:
Hires-Master und Special Basics: 85 neue blitzschnelle Grafikbefehle/ zwei ultimative Tools für Intros und Level-Screens/ Picture-Tool V1.0: klaut Bilder und Fonts aus fremden Programmen

SH 94:
Alan V7.3: komfortable Grafikerweiterung/ Big Pic: neun Scroll-Screens für Spiele/ Sprite-Edit: 32 Sprites für Action und Animation

Drucker

SH 72:
Publish 64: DTP-Einstieg/ Topprint: Druckt Briefköpfe, Schilder und Grußkarten/ Test: Drucker unter 1.000 DM

SH 88:
Giga-Publish: komplettes DTP-Paket/ Tips und Raffinessen zu 24-Nadel-Druckern/ Typwriter: Drucker füllt Formulare aus

C 128

SH 58:
Übersichtliche Buchhaltung zuhause/Professionelle Diagramme

SH 64: Anwendungen:
Amerikanisches Journal/ Grundlagen: CP/ M, das dritte Betriebs-system/ VDC-Grafik: Vorhang auf für hohe Auflösung

SH 76:
Music Master: Professionelle Datenbank zur LP- und CD-Sammlung/ Prüfungsaufgaben: Idealer Helfer vor jedem Examen

SH 82:
Mini-Micro: Kopiert 1571-Disketten zur 1581/ Pro-Book 128: Komfortable Datenbank für den Bücherwurm

SH 89:
DOS-Copy: Kabellose Datenüberftragung vom oder zum PC per Floppy 1571/ Codiman: Profi-Disk-Management/ Master-Basic: 51 neue Anweisungen und 25 Funktionen

SH 95:
Paint R.O.I.A.L.; Zeichenprogramm der Superlative/ Mini-dBase V2.1: relative Profi-Datenbank/ Daten-Grafik: aus Zahlen werden Diagramme

Spiele

SH 2: Top Spiele
10 Game-Trainer und 2 Longplays/ Katakis-System: Spiele programmieren wie die Profis/ Tolle Tips für höhere Level und Unsterblichkeit

SH 3: Top Spiele
Action Jump'n Run Logik/ Tips, Tricks & Tools

SH 6: Top Spiele
Perfect Symetrie: Das elektronische und raffinierte Puzzlespiel der Extraklasse/ Star Brain: An dieser genialen Mischung aus Action und Tetris kommt keiner vorbei Genloc: Erleben Sie die Abenteuer um Dr. Klein in diesem spielbaren Demo/ Ultima-Atlas: Mit den Karten zu Ultima 2 reisen Sie sicher und schnell durch die Lande von Lord British

SH 54:
15 tolle Spiele auf Diskette/ Der Sieger unseres Programmierwettbewerbs: Crillion II/ Ein Cracker packt aus: Das ewige Leben bei kommerziellen Spielen

SH 60: Adventures
8 spannende Abenteuerspiele/ 2 Komplettlösungen und viele Tips, Adventures selber programmieren

SH 61:
20 heiße Super-Games für Joystick-Akrobaten/ Cheat-Modi und Trainer POKEs zu über 20 Profi-Spielen/ Krieg der Kerne: Grundlagen Spieleprogrammierung

SH 66:
15 Top-Spiele mit Action und Strategie/ Mondlandung: Verblüffend echte Simulation und Super-Grafik/ High-Score-Knacker: Tips & Tricks zu Action-Games

SH 73:
Action bis Adventure: 10 Spiele zum Kampf gegen Fabelwesen/ Preview/ Tips & Tricks/ Kurse/ Game Basic/ Mission II/ W.P.Tennis II/ Omnibus GmbH/ Mic's Push'em

SH 79:
25 starke Mega-Games/ Exis: Gefährlicher Satelliten-schleppdienst/ Tips & Spielelösungen

SH 85:
11 Super-Games für stählerne Nerven/ Datenagent 00X: Noch 12 Stunden bis zum Weltuntergang/ Kick'n kill: Irrwitzigees Jump-and-Run-Spiel für Joystick-Akrobaten

SH 90:
10 Super-Games für Joystick-Künstler/ Cyborg 2900: 3D-Adventure hautnah erlebt/ Boulder Dash Construction Tool: neue Levels für den Game-Klassiker

SH 97:
11 brandneue Spiele auf Diskette/ Rock'n'Roll-Fahnder: Zoff im Piraten-sender/ Cheeky Twins: irre Diamantenjagd im Labyrinth der Monster

Nur noch hier erhältlich!
Jetzt sofort bestellen - per Post oder FAX !

Ordnung im eigenen Archiv für DM 14.-

Hier war leider jemand schneller, doch null Problem: Einfach schreiben und bestellen bei 64er -Magazin Leserservice, D-74170 Neckarsulm. Telefon 07132/969-185, FAX: 07132/969-190

BESTELLCOUPON

Ich bestelle folgende 64er Sonderhefte:

	SH-Nr.	SH-Nr.	SH-Nr.	
____Sonderhefte mit Diskette je 10,- DM				DM
____Sonderhefte"128er" je 10,- DM				DM

Ich bestelle______Sammelbox(en) ____DM
zum Preis von je DM 14,- **Gesamtbetrag** ____DM

Ich bezahle den Gesamtbetrag zzg.6,-DM Versandkosten
❑ nach Erhalt der Rechnung. ❑ per Scheck anbei

Vorname,Name

Straße, Hausnummer

PLZ, Wohnort

Datum / Unterschrift

Schicken Sie bitte den ausgefüllten Bestellcoupon an:
64er-Magazin Leserservice, D-74170 Neckarsulm, Telefon: 07132/969-185
oder bequem per Telefax: 07132/969-190

```
                    LDA #$00
POKE 53280,0
                    STA $d020
```

Die schematische Darstellung des POKE-Befehls in Assembler - der Basic-Befehl wird durch zwei Prozessor-Anweisungen ersetzt

findet sich ab 55296 (hex. $d800) und läßt sich analog zur Textausgabe beschreiben.

Für diese Aufgabe können Sie eine gesonderte Subroutine schreiben und sie aus dem eigentlichen Programm aufrufen. Dazu entwickeln Sie ein Programm und legen es an die Adresse $2000. Diese Aufgabe dürfte mit dem bisher gelernten Stoff nicht schwierig sein...

Wenn das kleine Programm fertig ist, verlassen Sie SMON. Da unser Testprogramm zur Textausgabe noch im Speicher steht, können Sie nun beide Routinen nacheinander aufrufen. Dazu geben Sie folgende Befehle im Direktmodus ein:

SYS 4096
SYS 8192

Nun müßte der das Wort String in der gewählten Farbe erscheinen. Wesentlich eleganter wäre es, die beiden Teilprogramme mit einem SYS-Befehl zu starten. Dazu könnten Sie die zweite Routine direkt nach dem Branch-Befehl an das Programm anhängen.

Eine andere Möglichkeit ist der Aufruf des zweiten Programms als Subroutine ähnlich wie in Basic mit dem Befehl GOSUB bzw. mit dem zweiten SYS (s.o.).

Um aus einem Maschinenprogramm eine Subroutine aufzurufen, benutzt man den JSR-Befehl. Das Assemblerlisting 5 zeigt die komplette Textausgabe mit einer Subroutine, die die Buchstaben weiß einfärbt. Hinter JSR befindet sich die Adresse, zu der der Prozessor springen soll. Am Ende der Subroutine muß immer ein RTS stehen, damit der Prozessor weiß, wann er vom Unterprogramm zurückkkehren muß.

Optimale Schleifen

Natürlich ist mit Listing 5 noch nicht das Optimum an Eleganz und Speicherausnutzung erreicht. Es geht noch kürzer. Listing 6 zeigt eine verbesserte Variante der Farbroutine. Das Setzen der Farbregister wurde in die Text-Routine einbezogen.

Natürlich können Sie sich auch eine gesonderte Farbtabelle anlegen und jedem Zeichen den gewünschten Farbton zuweisen. Ihrer Experimentier-Freude sind jetzt keine Grenzen gesetzt. Sie können z.B. Bildschirmausgaben in Assembler übertragen und Sie aus Ihren Basic-Programmen aufrufen.

Wechsel zwischen den Registern

Wer die letzten Abschnitte sorgfältig studiert hat, wird sich sicher noch an die beiden Befehle *DEX* und *DEY* erinnern. Natürlich gibt des auch zwei Anweisungen, die das X- und Y-Register um einen Zähler erhöhen. Sie heißen *INY* und *INX*. Dabei steht "IN" für incrementiere oder erhöhe das entsprechende Register um den Wert 1.

Wie Sie sicher schon gemerkt haben, gibt es bei der Verwendung der Befehle viele Parallelen. Bei den Decrement- und Increment-Anweisung ist das so ähnlich. Der Akku kann aber nicht wie ddas X- und Y-Register verändert werden, sondern nur eine Speicherstelle. *INC $D020* erhöht den Inhalt der Speicherzelle $d020 (dez. 53280) und *DEC $D020* zieht den Wert 1 vom Inhalt ab. Wer den Wert im Akku also auf- oder abzählen will, kann dessen Inhalt mit einem STA-Befehl in eine unbenutzte Speicherstelle schreiben und verändern.

STA $1000
INC $1000
LDA $1000

Dann wird das Ergebnis mit dem LDA-Befehl wieder in den Akku gehievt.

Natürlich verschlingt diese Methode Speicherplatz und schluckt gewaltig Rechenzeit. Eine bessere Lösung ist der Transfer zwischen den Registern. Dazu stellt der Prozessor des C 64 folgende Befehle bereit:

Befehl	Funktion
TAY	Akku nach Y schieben
TAX	Akku nach X schieben
TYA	Y in den Akku schieben
TXA	X in den Akku schieben

Zum Manipulieren des Akku-Inhalt nutzen wir dann folgende Routine:

```
TAY     ;akku nach y
INY     ;y=y+1
TYA     ;y in den akku
```

Eine weitere Möglichkeit den Akku-Inhalt zu verändern sind die arithmetischen Befehle des C-64-Prozessors. Dazu kommen wir im nächsten Monat. Dann beschäftigen wir uns weiter mit Schleifen und wollen die 8-Bit-Grenze der Register sprengen, neue Branch-Befehle kennenlernen und eine wenig rechnen.

Jörn-Erik Burkert

Einstieg in Assembler, erste Befehle, einfache Schleifen
Teil 2: Adressierungsarten, Branch-Befehle, Arithmetik
Teil 3: Weitere Programmiertricks, Betriebssystem und Praxis-Anwendung

Listing 1: Listing 2 als Maschinensprache-Version - diassembliert mit SMON

```
        9            40
1002    A2 27        LDX #27
1004    9D 00 04     STA 0400,X
1007    CA           DEX
1008    10 FA        BPL 1004
100A    60           RTS
```

Listing 2:Zeichensetzen mit einer FOR-NEXT-Schleife

```
10 A=64
20 FOR X=0TO39
30 : POKE1024+X,A
40 NEXT
50 END
```

Listing 3:Unterschiedliche Zeichen mit FOR-NEXT und READ auf denBildschirm bringen

```
10 FOR X=0TO3
20 : READ A
30 :  POKE1024+X,A
40 NEXT
50 DATA 20,5,19,20
```

Listing 4: Textausgabe in Assembler mit Hilfe einer Schleife

```
1000    A0 03        LDY #$03
1002    B9 0C 10     LDA $100C,Y
1005    99 00 04     STA 0400,Y
1008    88           DEY
1009    10 F7        BPL 1002
100B    60           RTS
100C    14           ***
100D    05 13        ORA
100F    14           ***
```

Listing 5: Textausgabe und Subroutine zum Setzen des Farb-RAMs (als Assembler-Quelltext)

```
start           ldy #$03
schleife1               lda text,y
                sta $0400,y
                dey
                bpl schleife1
                jsr farbe
                rts
farbe           ldy #$03
                lda #$01
schleife2               sta $d800,y
                dey
                bpl schleife 2
                rts
text     .byte  $14,$05,$13,$14
```

Listing 6: Listing 5 in komprimierter und optimierter Form (als Assembler-Quelltext)

```
start           ldx #$03
schleife        lda text,x
                sta $0400,x
                lda #$01
                sta $d800,x
                dex
                bpl schleife
                rts
text     .byte  $14,$05,$13,$14
```

Highlights

Programm-Service-Disk

64'er 7/95

Diskette Seite A

Workshop Dateiverwaltung: Reldat V2.0
Eingabe-Routinen
Geometrie-Profi
TabCalc 128
Quicksort 64/128 in Assembler
Geo-Funktion

Diskette Seite B

Top-Game auf Disk:
Quadrant – Strategie
im Tetris-Look



Kleinanzeigen-Auftrag für den COMPUTER-MARKT

Bitte veröffentlichen Sie in der nächsterreichbaren Ausgabe von 64'er den folgenden Kleinanzeigentext unter der Rubrik

________________ (z.B. C 64, C 128, VC 20, Software, Zubehör, Verschiedenes).

Meine Anzeige ist eine ☐ **private Kleinanzeige** (5 Zeilen mit je 32 Buchstaben, maximal 160 Zeichen)

☐ DM 5,– liegen ☐ bar ☐ als Scheck bei. **Bitte keine Briefmarken!**

Bezahlung über Postscheckkonto nicht mehr möglich.

Meine Anzeige ist eine ☐ **gewerbliche Kleinanzeige für DM 12,–** (zzgl.MwSt.) **je Druckzeile**

Anschrift:

64'er
Computermarkt
MagnaMedia Verlag AG
Postfach 1304
85531 Haar

Absender:

Name/Vorname

Straße

PLZ/Ort

Telefon

Bei Angeboten: Ich bestätige, daß ich alle Rechte an den angebotenen Sachen besitze

Datum: Unterschrift:



Impressum

Chefredakteur: Wolfram Höfler
Stellv. Chefredakteur: Harald Beiler (bl), verantwortlich für den redaktionellen Teil
Textchef: Jens Maasberg
Chef vom Dienst: Dorothea Ziebarth (dz)
Redaktion: Jörn-Erik Burkert (lb)
Mitarbeiter dieser Ausgabe: Denis Döhler Diethelm Kretschmann, Christian Schäffner, Andre Marth
Redaktionsassistenz: Sylvia Simon (sd)

So erreichen Sie die Redaktion:
Tel. 089/4613-202, Fax 089/4613-433, Btx *64064#

Manuskripteinsendungen: Manuskripte und Programmlistings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten worden sein, so muß das angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in den von der MagnaMedia Verlag AG herausgegebenen Publikationen und zur Vervielfältigung der Programmlistings auf Datenträgern. Mit Einsendung von Bauanleitungen gibt der Einsender die Zustimmung zum Abdruck in von MagnaMedia Verlag AG verlegten Publikationen und dazu, daß die MagnaMedia Verlag AG Geräte und Bauteile nach der Bauanleitung herstellen läßt und vertreibt oder durch Dritte vertreiben läßt. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

Layout: Uschi Böcker
DTP-Operator: Dorothea Voss, Hans-Dieter Schimank
Titellayout: Uschi Böcker
Computergrafik: Alexander Gerhardt
Fotografie: Roland Müller

Anzeigenverkaufsleiter: Regine Schmidt (828), verantwortlich für den Anzeigenteil
Anzeigenverwaltung / Disposition: Susanne Schröder (853)

So erreichen Sie die Anzeigenabteilung
Tel. 089/4613-962, Telefax 089/4613-394

Anzeigenpreise: Es gilt die Preisliste Nr. 3 vom 01. 01. 1995
International Account Manager: Kurt Skupin (352)
Assistenz: Michelle Berner (360), Fax 089/4613-775

Anzeigen-Auslandsvertretungen:
Großbritannien: Smyth International, London, Tel. 0044-83140-5058, Fax 0044-81341-9602
Frankreich: Ad Presse International S.a.r.l. 34, rue de Camille Pelletan, F-92300 Levallois-Perret, Tel. 0033-1-47317530, Fax 0033-1-47317507
Taiwan: Acer TWP Co., Taipei, Tel. 008862-713-6959, Fax 008862-715-1950
Italien: Medias International, Mariano, Tel. 0039-31-751494, Fax 0039-31-751482
Holland: Insight Media, Laren, Tel. 0031-2153-12042, Fax 0031-2153-10572
Israel: Baruch Schaefer, Holon, Tel. 00972-3-556-2256, Fax 00972-3-556-6944
Korea: Young Media Inc, Seoul, Tel. 00822-765-4819, Fax 00822-757-5789
Hongkong: The Third Wave (H.K.) Ltd., Tel 00952-7640989, Fax 00852-7643857
Erscheinungsweise: monatlich (zwölf Ausgaben im Jahr)

Abonnement-Service:
64'er Aboservice
74168 Neckarsulm, Tel.: 07132/959-242,
Fax: 07132/959-244
Einzelheft: DM 9,80
Jahresabonnement Inland
(12 Ausgaben): DM 105,–
(inkl. MwSt., Versand und Zustellgebühr)
Jahresabonnement Ausland: DM 129,–
(Luftpost auf Anfrage)
Österreich: DSB-Aboservice GmbH,
Arenbergstr. 33, A-5020 Salzburg, Tel.: 0662/643866,
Jahresabonnementpreis: öS 864
Schweiz: Aboverwaltungs AG, Sägestr. 14,
CH-5600 Lenzburg, Tel.: 064/519131,
Jahresabonnementpreis: sfr. 105.–
Nachbestellung Einzelhefte:
64'er Leserservice
Heiner-Fleischmann-Str. 2, 74172 Neckarsulm
Tel. 07132/969-181 – Fax 07132/969-190
(Heftpreis + DM 6,– Versandpauschale)

Leitung Herstellung: Klaus Buck (180)
Technik: Sycom Druckvorstufen GmbH, Hans-Pinsel-Straße 2, 85540 Haar
Druck: Druckerei E. Schwend GmbH & Co. KG, Schmollerstraße 31, 74523 Schwäbisch Hall

Haftung: Für den Fall, daß im 64'er unzutreffende Informationen oder in veröffentlichten Programmen oder Schaltungen Fehler enthalten sein sollten, kommt eine Haftung nur bei grober Fahrlässigkeit des Verlages oder seiner Mitarbeiter in Betracht.
Vertriebsleitung: Benno Gaab
Sonderdruck-Dienst: Alle in dieser Ausgabe erschienenen Beiträge können für Werbezwecke in Form von Sonderdrucken hergestellt werden.
Anfragen an:
Klaus Buck,
Tel. 089/4613-180,
Telefax: 089/4613-232

1995 MagnaMedia Verlag Aktiengesellschaft

Vorstand: Carl-Franz von Quadt (Vors.), Kenneth Clifford, Eduard Unzeitig

Verlagsleiter: Wolfram Höfler

Anschrift des Verlages:
MagnaMedia Verlag Aktiengesellschaft,
Postfach 1304, 85531 Haar bei München,
Telefon 089/4613-0, Telex 522052, Telefax 089/4613-100

Diese Zeitschrift ist auf chlorfreiem Papier mit einem Altpapieranteil von 100% gedruckt. Die Druckfarben sind schwermetallfrei.

64'er COMPUTER-MARKT

Wollen Sie einen gebrauchten Computer verkaufen oder erwerben? Suchen Sie Zubehör? Haben Sie Software anzubieten oder suchen Sie Programme oder Verbindungen? Der COMPUTER-MARKT von »64'er« bietet allen Computerfans die Gelegenheit, für nur 5,- DM eine private Kleinanzeige mit bis zu 4 Zeilen Text in der Rubrik Ihrer Wahl aufzugeben. Und so kommt Ihre private Kleinanzeige in den COMPUTER-MARKT der **September-Ausgabe** (erscheint am 25.08.95): Schicken Sie Ihren Anzeigentext bis 20. Juli (Eingangsdatum beim Verlag) an »64'er«. Später eingehende Aufträge werden in der **Oktober-Ausgabe** (erscheint am 29.09.95) veröffentlicht.

Am besten verwenden Sie dazu den vorbereiteten Coupon im Heft.

Bitte beachten Sie: Ihr Anzeigentext darf maximal 4 Zeilen mit je 40 Buchstaben betragen.

Schicken Sie uns DM 5,- als Scheck oder in Bargeld. Der Verlag behält sich die Veröffentlichung längerer Texte vor. Kleinanzeigen, die entsprechend gekennzeichnet sind, oder deren Text auf eine gewerbliche Tätigkeit schließen läßt, werden in der Rubrik »Gewerbliche Kleinanzeigen« zum Preis von DM 12,- je Zeile Text veröffentlicht.

Private Kleinanzeigen

COMMODORE 64

Verk C 64 II, 1541I, Final Cartridge 3, Module, Disketten u. Box, Joystick, 64'er-Einsteiger-Kurs, alles 100% OK, 350 DM; Tel. 05432/1517 (Carsten)

Biete C 64 Tastaturen ohne Platinen für 20 DM. Final Chesscard NP 150 DM für nur 35 DM. Diverse C 64 Disketten mit Prg's. und orig. Spielen. (Günstig) Tel. 05776/1071

Verk. C 64 + Floppy + Disketten + Bücher + viel Zubehör, 270 DM, Tel. 089/5237667 (kann per Post geliefert werden)

Verk.: (Einz./zus.) C 64, 1541, Seikosha 1000, FC 3, Quickjoy, Geos, 64-Buch, 64-Hefte, Leedisks, Diskbox, Preise nach Absprache, (billig) 03943/602573, Jens, öfters versuchen

Suche neuwertigen Drucker (24 N), wenn vorhanden mit Einzelblatteinzug und CMD-Smart M. (100% OK) Tel. 0043-7727-2038 ab 13 Uhr

IC für C 64 und 1541, 2 x 6522, 6502, 6510, 6526 A, 251913-01, 251968-03, 901229-05, 27128, NP. ca. 190 DM für 90,– DM zu verk. (nur kompl.) Tel. 08677/63973

Alte 64'er-Hefte: 5-12/84, Jahrg. 85, 86, Sonderh. PEEK + POKE, div. Happy-Comp., DB-Bücher: Maschinenspr., 64 Intern, sow. Drucker Riteman C+ und 1541 abzug. Tel. 0531/612239

Verk. C 64 Brotkasten 70,– DM, Floppy 1541 II 70 DM, Drucker MPS-1230, 100 DM, Geos 2.0, 50 DM, viele Software und Spiele, Bücher 64'er Hefte. Tel. 0355/538906

C 64 + 1541 + Jiffy-DOS eingeb. 195 DM. C 64 + 1541 + SpeedDOS+ eingeb. 210 DM. C 64 + 1541 + Dolphin-DOS einbeb. 260 DM. Expan.-Wei. 4 oder 5 Fach je 50 DM. Seikosha 1900 180 DM. Tel./Fax/Btx 02303/80916

Verk: C 64 II, Floppy 1541, Zubehör, 1 x Steckplatzerw., 1 Diskbooster-Modul, 3 Joyst., 1 Userportresetsch., gr. C 64-Hanbuch, C 64'er, 08/94-07/94 zus. für 300,– DM. Tel. 0351/2847872

Verk. 1541/2 mit Trackdisplay, elektr. Disklocher, Parallelkabel + Burstnibbler + Rapidcopy, ohne Netzteil kpl 100,– DM, weiteres Zubehör w. Sys. Auflösung z. verk. Liste anfordern. Tel. 0177/2037520

COMMODORE 128

C 128 + Disketten + Bücher + viel Zubehör + evtl. Monitor (Farbe) 350 DM (mit Monitor); 300 DM (ohne Monitor). Tel. 089/5237667

C 128DB, Geos 2.5, Geo-File, Geos PD, 1571 II, GeoRAM 512 K, Monitor 1081, Software z.B.: Ultima 1-6, Zak McKracken, Dragon Wars, Centronics-Interface, alles VHS. Tel. 07083/4462

C 128 + 1571 + Jiffy-DOS eingeb. 295 DM. C 128 + 1570 210 DM. Fl. 1581 + Jiffy-DOS eingeb. 250 DM. BTX-Modul II 60 DM. Modem 1200 bps 30 DM, 2400 bps 75 DM. Tel./Fax/Btx 02303/80916

Wichtiger Hinweis:
Zur Bezahlung von Kleinanzeigen werden weiterhin keine Briefmarken angenommen.

SOFTWARE

Suche das orig. SSI Rollenspiel „Champions of Krynn". Zahle bis zu 50 DM für 100% lauffähiges Spiel mit Anleitung und anderem Zubehör. Tel. 02181/44477 (Thomas)

Suche: Geos 2.5, Geopublish, Geo-File, Geocalc, Geochart --- nur Originalverpackt, Deutsche Versionen, komplett mit Handbuch für je 20-50 DM --- Tel. 0251/25662

ZUBEHÖR

Verk. Pagefox Modul 130,–, Handyscan 150,–, DIN A3 Plotter C 64/128 200,–, Eprombr. Quickbait 2 50,–, Epromkarte 1 MB Leer 50,–, Original Disketten und Zubeh.; Tel. 0265/51796 Abends

Zu verk.: 64'er Magazine mit Disk, Magic Disk, Bücher, Software, IC's, Hardware, Zubehör, Liste od. Info bei: D. Rivola, Am Anger 32, 86316 Friedberg, Tel. 0821/782913

Reu 1764/512K 175 DM. GeoRAM 512K 100 DM. 192 K Eprompl. f. voll Best. 75 DM, Prospeed 128 u. Dolphin-DOS, je 100 DM. Proport.-Maus 1351 40 DM. IC-Tester 50 DM. Tel./Fax/Btx 02303/80916

Verkaufe: 4 MHz-Karte 100,–; Lightpen 30,–; Btx-Modul 50,–; Modulspiel 20,–; Drucker Seikosha 120 VC 130,–; u.v.m. Anfragen unter: 034721/22506 (ab 18 Uhr) Daniel G.

Neue Netzteile: Für C 128 40,–, für Floppy 1541-2 und 1581 50,–, Power-Netzteil (4,3 A) für C 64 50,–, neue Tastatur für C 64 30,–, Tel. und Btx 0201/743653

Philips TV-Tuner (macht aus dem Monitor einen Fernseher) für 140 DM VHB. Simons Basic Modul, deutsche Anleitung, für 15 DM VHB. Manfred Weiffen, Tel. 06742/2197

VERSCHIEDENES

Verkaufe 8502 und 6526A zus. 30 DM + Peeks & Pokes 128, Tips & Tricks 128, Das große Basic-Buch für 70 DM, Stefan Voswinkel, Hauptstr. 123, 37308 Geisleden

Suche C 64 II und/oder 1541 II ganz oder defekt auch Monitore und Drucker sind willkommen. B. Westphal, Augustenstr. 71, 06366 Köthen/Anh.

Verschenke versch. Software (Disk) mit Anleitung (alles Originale) sowie versch. 64'er Hefte. Liste mit 1,– DM Rückporto anfordern: A. Brandt, Bröderhausener Str. 32, 32549 Bad Oeynhausen

Verkaufe C 16+4 mit Zubehör z.B. Floppy 1551, Datas, Spiele auf Band, 2. Tastatur zum Ausschl. etc. gegen Gebot. Tel/Btx 0202/63138

Verk. Monitor 1802, Tastatur 64, versch. Bücher, 64'er Magazine, versch. Zubehör, Preis auf Anfrage bei: Markus Zajac, Weisgutstr. 16, 33106 Paderborn, Tel. 05254/68177

Sammler! 64'er zu verkaufen! 4/84-12/84 2/85-5/85, 7/85-12/85, 1/86-3/86, 9/86-11/86 Sonderheft 21 Assembler, bitte melden: Maik Wandtke, Pf 161022, 18023 Rostock

Biete: Orig. Handyscanner 64 (Scantronic, NP 498,–) für 250,–. Tel. 09776/9866 (ab 18 Uhr)

64'er Zeitschriften 1984 teilw., 1985-89 kompl., Datasette, Monitor monochr., C 64 def. und 1541 def. gegen Gebot. Tel. 0821/663495

Gewerbl. Kleinanzeigen

Software, Telespiele u. Zubehör, Preisliste, Tel. 06447/285

C 64/128/128D ab 80,–/100,–/150,– DM, Floppy 1541 ab 80,–, Floppy 1571 ab 130,–, Monitor ab 80,–, Drucker ab 80,–, versch. Soft.- + Hardware u.v.m.; Liste anfordern: Tel. 0871/63829, Fax: 64995 sol. Vorrat reicht

Wichtige Hinweise für alle Kleinanzeigeninserenten

- Kleinanzeigenaufträge **ohne Absenderangabe** sowie Anzeigentexte unter **Postlagernummer** können leider **nicht** veröffentlicht werden.
- Bitte verwenden Sie für Ihren Auftrag das Formular auf Seite 20.
- Zur Bezahlung von Kleinanzeigen werden **keine Fremdwährungen** mehr angenommen.
- Bitte achten Sie auch darauf, daß Ihr Auftrag immer **vollständig ausgefüllt** ist (z.B. Unterschrift).

Grundlagen: Assembler-Programmierung

Maschinensprache ohne Hindernisse

Basic-Programmierer schalten den C 64 ein und können sofort loslegen. Wer sich auf Maschinensprache-Ebene begibt, braucht etliche Programmier-Werkzeuge. Wir zeigen Ihnen, welche Tools zur Entwicklung von Software in Maschinensprache zur Verfügung stehen.

Bevor es mit der Programmierung in Maschinensprache losgeht, muß man wissen, daß dem Prozessor eigentlich nur Befehle in Form von Hexadezimal-Werten vorgegeben werden. Damit die Eingabe der Assembler-Befehle in Hex-Kodierung nicht zur Sisyphusarbeit ausartet, wurden zahlreiche Tools für diese Aufgabe entwickelt. Dem Assembler-Programmierer bleiben also die Übersetzung der Befehle mit Hilfe von Tabellen und die Übertragung per POKE-Befehl erspart. Man unterscheidet bei den Maschinensprache-Tools zwischen zwei Gruppen:

1. Maschinensprache-Monitor
2. Assembler.

Beide verstehen sogenannte Memonics, die die eigentlichen Maschinensprache-Befehle repräsentieren und übersetzen sie in ein lauffähiges Maschinenprogramm. Dieser Vorgang heißt Assemblierung.

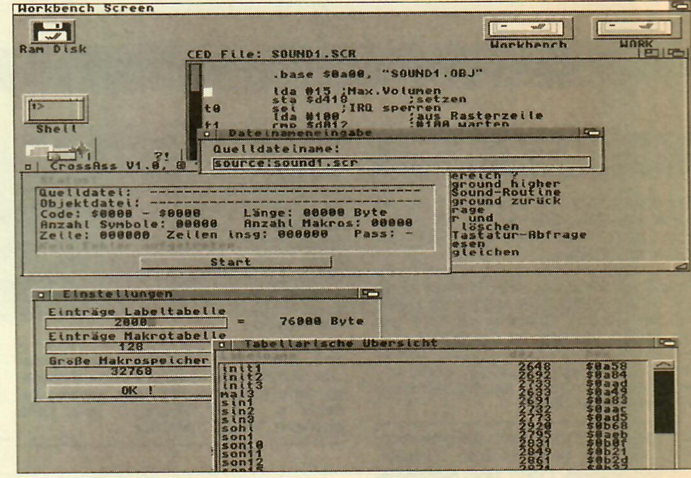
Der Maschinensprache-Monitor

Mit dem Maschinensprache-Monitor nimmt man den Speicher des C 64 unter die Lupe und kann die kodierten Befehle in Klartext übersetzen. Neben der Diassemblierung kann jeder gute Maschinensprache-Monitor (Monitor) auch die Inhalte der Speicherzellen zeigen, Befehle assemblieren und dem Programmierer die Möglichkeit eröffnen die Speicherstellen zu manipulieren. Zusätzliche Funktionen, wie Sprite- oder Zeichensatz-Optionen gehören nicht immer zum Lieferumfang.

Der Monitor ist zum Testen von Programmen und zur Jagd auf Bugs (Fehlersuche) ein sehr wertvolles Hilfsmittel. Viele Monitore haben deshalb einen integrierten Trace-Modus. In diesem Mode lassen sich im Programm verschiedene Haltepunkte setzen, an denen der Monitor das Programm unterbricht. Nun kann der Coder sein Programm und die Prozessor-Flags in aller Ruhe untersuchen. Per Befehl wird das Programm wieder in Gang gesetzt.

Mit einem Monitor kann man durchaus Programme schreiben, leider nicht so komfortabel wie mit einem Assembler. Die Befehle werden direkt in den Speicher übertragen. Möchte man Anweisungen einfügen, ist der gesamte Code im Speicher zu verschieben. Außerdem verarbeitet der Monitor keine Labels. Darum müssen bei Programm-Sprüngen die direkten Adressen hinter den JMP-/JSR-Befehlen stehen. Im Falle einer Programmsverschiebung ist die Anpassung der betreffenden Adressen notwendig.

Ein Monitor belegt immer einen Teil des Speichers, der sich dann nicht für eigene Programme nutzen läßt. Wird er überschrieben, kommt es zu 100 Prozent zum System-Crash. Besitzer eines Multifunktions-Moduls sind da schon ein wenig besser dran, denn viele dieser Hardware-Erweiterungen haben einen eingebauten Monitor. Außerdem haben sie den Vorteil, daß sie keinen Speicher im C 64 belegen.



Der "X-ASS-64" erlaubt die Entwicklung und Assemblierung von C-64-Programmen auf dem Amiga

```
C11A   A0 A2       LDY #A2
C11C   A1 C1       LDA (C1,X)
C11E   21 61       AND (61,X)
C120   84 86       STY  86
C122   E6 C6       INC  C6
C124   E0 C0       CPX #C0
C126   24 4C       BIT  4C
C128   20 90 B0    JSR B090
C12B   F0 30       BEQ C15D
C12D   D0 10       BNE C13F
C12F   50 70       BVC C1A1
C131   78          SEI
C132   00          BRK
---------------------------------
C133   18          CLC
C134   D8          CLD
C135   58          CLI
C136   B8          CLV
C137   CA          DEX
C138   88          DEY
R
  PC  SR AC XR YR SP  NV-BDIZC
 C00B B0 C2 00 00 F6  10110000
```

Der Maschinensprache-Monitor "SMON" ist ein ideales Utillity für die Fehlersuche in Assembler-Programmen

Der Assembler

Mit einem Assembler schreibt man umfangreiche Maschinensprache-Programme und übersetzt sie in lauffähigen Code. Dabei muß der Programmierer keine festen Adressen angeben und kann mit Labels arbeiten. Einzige Ausnahme ist die Start-Adresse des Programms.

Zusätzlich lassen sich komfortabel Texte bzw. Tabellen einfügen und Macros integrieren. Spätere Änderungen oder das Einfügen von Programmzeilen sind kein Problem, da der Assembler den Quellcode nicht sofort in den Speicher überträgt. Dieser Vorgang erfolgt gesondert mit einem speziellen Befehl. Ein weiterer großer Vorteil des Assemblers, zeigt sich in der Möglichkeit die Quelltexte zu dokumentieren. Der Assembler ignoriert diese Infos beim Übersetzen, sichert aber auf Wunsch den kompletten Quelltext.

```
  VisAss V4.0     (c) 1992   by Vislogic
info    disk    prefs    ass    edit    extra
 £la makro=writepos+1
 £la savecall=makro+1
 £la asciitab=savecall+1
;
; *** makros
;
 £md text.txtadr
          lda #<txtadr
          ldy #>txtadr
          jsr $ab1e
 £de
;
; *** basic teil
;
 £wo basline
 £wo 1992
 £by $9e
 £tx "2061"
basline:
 £by $00,$00,$00
AZ:44     ZA:$44dc  GL:1383    X:1   Y:20
```

Der "VIS-Ass" läßt keine Frage offen - die Programmierung in Maschinensprache wird zum Kinderspiel

Der Assembler wird wie ein Maschinensprache-Monitor als Software geladen und belegt komplett mit dem Quelltext einen großen Teil des C-64-Speichers. Wer also größere Programme schreiben möchte, muß sie meist aus Speicherplatz-Mangel in Teilprogramme splitten und sie später nacheinander zum Test in den Speicher laden. Eine andere Möglichkeit ist die Nutzung eines Cross-Assembler-Systems. Hier werden die C-64-Programme auf einem anderen Computer (z.B. zweiter C 64, Amiga oder PC) entwickelt und per Kabel in den Speicher des Testrechners geschaufelt und dort getestet.

Jörn-Erik Burkert

Der Monitor SMON

Wer noch keinen Maschinensprache-Monitor in seiner Programm-Sammlung hat, ist mit "SMON" bestens bedient. Mit Hilfe des Monitors können Sie den Kurs "Von Basic nach Assembler" verfolgen und die vorgestellten Beispiele nachvollziehen. Der Maschinensprache-Monitor belegt in dieser Version den Speicherbereich 49152 (hex. $c000) bis 53247 (hex. $cfff). In diesen Memory-Block dürfen Sie keine Programme laden oder assemblieren, sonst wird der Monitor zerstört und der C 64 kann abstürzen.

Die erweiterte Version des SMON ermöglicht die selbstständige Anpassung des Monitors an einen anderen Speicherbereiche. Außerdem können Sie mit der verbesserten Variante Disketten unter die Lupe nehmen, da ein Disketten-Monitor integriert ist und zusätzliche Befehle nutzen. Das Programm und andere Assembler-Tools finden Sie auf der 64'er-Sonderdiskette "Assemblerpaket" (s. Rubrik "Software-Klassiker").

Die einzelnen Befehle des Programms erwarten bei der Eingabe Parameter. Die Abkürzungen für die Parameter bedeuten dabei:

aa Anfangsadresse
ea Endadresse
za Zieladresse

Der Monitor wird mit

```
LOAD"SMON $c000",8,1
```

direkt in den Speicher geladen und mit SYS 49152 gestartet. Dann stehen folgende Befehle zur Verfügung.

Die SMON-Befehle

A aa
Assemblierung ab Adresse aa

B aa ea
Ausgabe des Speicherbereichs aa bis ea als Basic-Datas

C aaALT eaALT aaNEU eaNEU aaUM eaUM
Verschiebt den Bereich aaALT bis eaALT nach aaNEU bis eaNEU. Außerdem werden alle Adressen im Bereich aaUM bis eaUM dem neuen Speicherbereich angepaßt.

D aa ea
Disassembliert den Bereich von aa bis ea.

E
Rücksprung ins Basic und Aktivierung des Basic-ROMs.

F wert(e) aa ea
Sucht im Bereich aa bis ea nach HEX-Werten.

G aa
Ruft ein Maschinenprogramm auf, dessen Start-Adresse sich an aa befindet. Der Abschluß des Assembler-Programms mit dem BRK-Befehl sorgt für die Rückkehr zu SMON

I gerätenummer
Setzt das aktuelle Gerät für LOAD bzw. SAVE.

K aa ea
Listet im Bereich aa bis ea alle ASCII-Zeichen, die auch durch Überschreiben verändert werden können.

L name aa
Laden eines Programmes an die Adresse aa. Unterbleibt die Adressenangabe, lädt SMON automatisch an die Original-Startadresse.

M aa ea
Gibt den angegebenen Speicherbereich als HEX-Dump auf dem Bildschirm aus. Byte-Änderungen durch Überschreiben sind möglich.

O aa ae HEX-wert
Belegt den Speicherbereich aa bis ea mit dem vermerkten HEX-Wert.

P nummer
Setzt die Drucker-Adresse (Voreinstellung 4).

R
Zeigt die Prozessor-Register.

S name aa ea
Sichert einen Speicherbereich von aa bis ea unter "name".

W aa ea za
Verschiebt den Speicherbereich aa bis ea ohne Umrechnung nach za.

X
Springt zu Basic zurück, ändert aber Speicherstelle 1 nicht (s. E-Befehl).

dz
Rechnet die Dezimalzahl dz in entsprechende HEX-Zahl um.

$ hz
Rechnet HEX-Zahl hz in dezimalen Wert um. Ist die Zahl kleiner $100, wird zusätzlich der Binär-Wert angegeben.

% bz
Umrechnung der Binär-Zahl bz in entsprechenden HEX- und Dezimal-Wert.

V aaALT eaALT aaNEU
Vergleicht den Speicherbereich von aaALT bis eaALT mit dem Memory-Block ab aaNEU.

Glossar: Maschinensprache-Programmierung

Assemblierung: Übersetzung von Quelltexten (bestehen aus ☞Memonics, ☞Labeln und Tabelle) mit Hilfe eines Assemblers in lauffähigen Maschinencode.

Assembler: Hilfsprogramm zum Erstellen von Maschinensprache-Programmen. Der Assembler arbeitet mit Quell- oder ☞Source-Code, der per Befehl in lauffähige Programme übersetzt wird. Maschinenprogramme werden von Basic mit dem SYS-befehl gestartet.

Bekannte Assembler sind: VIS-Ass, Hypra-Ass oder Turbo-Ass.

Disassemblierung: Rückübersetzung von Assembler-Programmen mit Hilfe eines Maschinensprache-Monitors. Die kodierten Werte aus den Speicherzellen werden dabei in lesbare Befehle übersetzt. Eine weitere Möglichkeit der Rückübersetzung mit einem Reassembler. Er erzeugt lesebaren ☞Source-Code zur Weiterverarbeitung in einem ☞Assembler. Reassembler sind speziell auf einen Assembler-Type zugeschnitten.

Hex-Codes: Werte der C-64-Speicherzellen in ☞*hexadezimal*er Schreibweise. Sie können Daten oder Prozessor-Befehle widerspiegeln. Hex-Codes werden auch vom MSE verwendet, mit dessen Hilfe man Maschinenprogramme aus älteren 64'er-Ausgaben abtippen kann.

Hexadezimal: Zahlensystem das auf der Basis 16 aufbaut. Im Gegensatz zum dezimalen System wird nach der Ziffer 9 die Reihenfolge mit A, B C, D, E und F fortgesetzt. Nach der hexdezimalen Zahl F folgt "10".

Label: Marken die beim Schreiben von Maschinensprache-Programmen für Adressen gesetzt werden. Der Assembler überträgt diese Labels in feste Adressen, wenn er den Quelltext in ein lauffähiges Programm übersetzt.

Macro: Programmteil der sehr häufig gebraucht wird. In Assembler-Quelltexten wird er nur einmal definiert und automatisch beim Übersetzen des Maschinensprache-Programms eingefügt. Diese Methode verschlingt sehr viel Speicherplatz und ist deshalb durch Unterprogramme zu ersetzen.

Memonic: Lesbare Form der Maschinensprache. Die eignetlich ☞hexdezimal codierten Maschinensprache-Anweisungen werden als Befehl dargestellt (z.B. LDA oder STY).

Source-Code: Ein Maschinenprogramm als Quelltext mit lesbaren Maschinensprache-Befehlen (☞Memonics); Anweisungen und ☞*Labels.* Der Source-Code wir in einem ☞Assembler eingeben und kann dort in lauffähigen Maschinencode überstezt werden. Der Quelltext kann auf Diskette gesichert und bei Bedarf geändert werden.

Grundlagen: Eingabe-Routinen

Input-Know-How

Programmabläufe steuert man entweder per Keyboard, Joystick oder Maus. Eingabe-Routinen zu programmieren ist gar nicht so schwer. Treffende Beispiele helfen Ihnen dabei auf die Sprünge.

Unsere Tips sollen Ihnen Denkanstöße vermitteln, damit Sie Ihre selbstentwickelte Software auf Vordermann bringen. Die vorgestellten Beispiele lassen sich beliebig erweitern und modifizieren.

Die Tastatur

Um Eingaben per Keyboard zu realisieren stellt der C 64 zwei Standard-Lösungen zur Verfügung: INPUT- und GET-Funktion.

Der INPUT-Befehl ist hinterlistig: er zerstört bei Fehleingaben die Bildschirmmaske und fängt falsche Werte nicht ab. Mit GET umgeht man diese Fallen. Hier werden Tastatur-Eingaben zeichenweise eingelesen und lassen sich anschließend auswerten. In Listing 1 finden Sie eine Eingabe-Routine mit GET in Basic, Listing 2 bringt die Lösung in Maschinensprache.

Beide Programme fragen die Buchstaben von A bis Z ab. Alle anderen Tasten werden ignoriert (Ausnahme: RUN/STOP und RESTORE). Wollen Sie Ziffern abfragen, müssen Sie die Grenzen (64 bis 90) an die ASCII-Codes der Ziffern-Tasten (48 bis 57) anpassen. Die Cursor-Anzeige, Check auf ein Komma und die RETURN-TASTE als Abschluß der Eingabe sind als zusätzliche Features denkbar. Außerdem muß man die eingebenen Zeichen für die Weiterverarbeitung in Strings bzw. Zahlen wandeln. Um einen String zu erzeugen, fügen Sie nur die Zeile

```
55 N$=N$+T$
```

in Listing 1 ein. Wenn Sie das Programm nach einigen Eingaben mit der STOP-Taste unterbrechen, können Sie sich im Direktmodus mit dem Befehl

```
PRINT N$
```

den erzeugten String N$ zeigen lassen. In der Assembler-Version (Listing 2) sind die eingebenen Zeichen aber in einem String-Feld zu deponieren.

Die Eingabe von Zahlenwerten ist da schon komplizierter, denn hier ist die Umwandlung des Strings in numerische Werte notwendig. Dazu dient die Routine in Listing 3.

Sollten sich im String Zahlen mit Nachkommastellen befinden, muß man noch einen Check auf das Komma in Listing 3 eingefügen und die Umwandlung des String-Inhalts modifizieren. Listing 4 enthält eine Routine, die diese Aufgabe übernimmt.

Für komfortable Bedienung sind Tastenkombinationen sehr nützlich (z.B. CTRL+L). Diese Abfrage erledigt man am besten per Tastatur-Matrix. Dazu muß man wissen, daß das Keyboard mit dem CIA-Baustein verbunden ist und die Kontrolle sich auch über die CIA-Register 56320 und 56321 (hex. $dc00/$dc01) steuern läßt. Beide Adressen dienen eigentlich der Abfrage beider Joystick-Ports. Den Aufbau der 8x8 Felder großen Tastatur-Matrix finden Sie in Tabelle 1.

Um korrekte Eingabe-Ergebnisse mit der Tastatur-Matrix zu erzielen, empfiehlt sich Assembler. Listing 5 zeigt die Abfrage per Tastatur-Matrix. Unser Beispiel testet die CTRL-Taste. Dazu verbiegen wir den IRQ-Vektor auf unsere Tastatur-Routine. Im Programmteil "IRQ" wählen wir die Tasten der Spalte 7 in der Tastatur-Matrix, durch Eintrag des Wertes 127 (binär %01111111) ins CIA-Registers 56320 (hex.$dc00). Danach wird Register 56321 (hex. $dc01) gelesen. Die folgende Maskierung sorgt dafür, daß alle eventuell gesetzen Bits, mit Ausnahme des fünften, gelöscht werden. Wird ein Druck auf die CTRL-Taste registriert (Wert im Akku ist Null), erhöht der nächste Befehl das Register für die Farbe des Bildschirm-Rahmens und springt dann zur Standard-IRQ-Routine zurück. Wenn nicht, wird der INC-Befehl ausgelassen. Natürlich sind Kombinationen oder getrennte Abfragen der beiden SHIFT-Tasten möglich. Dazu sind die entsprechenden Bits beim Beschreiben von $dc00 bzw. beim Lesen von $dc01 zu setzen.

```
L
10 X%=0:Y%=0:J%=0:REM TREIBERVARIABLEN
20 IFA=0THENA=1:LOAD"MAUS.DRV",8,1
30 SYS 52887
READY.
PRINTX%
 174

READY.
PRINTY%
 70

READY.
PRINTB%
 0

READY.
```

Der Maustreiber auf der Diskette zum Heft initialisert ein Sprite und gibt die Koordinaten des Zeigers ständig in Basic-Variablen zurück

Der Joystick

Wie bereits zuvor beschrieben, lassen sich die Joystick-Ports über die Register 56320 (hex $dc00) bzw. 56321 ($dc01) abfragen. Bei beiden Speicherstellen sind die untersten fünf Bits für die Abfrage der Steuerhebel-Bewegungen und den Feuerknopf interessant.

Beim Lesen der Joystickports wird getestet, ob eines der fünf Bits gelöscht ist. In Tabelle 2 finden Sie deren Bedeutung. Natürlich sind auch Kombinationen möglich (z.B. rechts-oben).

Basic-Programmierer bevorzugen die Lösung von Listing 6. Nach der Abfrage stehen in der Variablen J die Werte 1 (nichts), 2 (oben), 3 (unten), 4 (links), 5 (rechts) oder 6 (Feuer). In Zeile 30 werden die entsprechenden Sprungadressen für die Subroutinen angegeben. Auf der Diskette zum Heft finden Sie einen Joysticktreiber. Er arbeitet komplett im Interrupt und übergibt die jeweilige Richtung an die Variable J. Dabei muß Ihr Listing in jedem Fall mit dieser Befehlsfolge beginnen:

Programme auf Diskette

Auch die Beispiele zu diesem Artikel finden Sie auf der Diskette zum Heft. Die Basic-Programme werden wie gewohnt geladen und gestartet. Die Maschinensprache-Beispiele finden Sie ab Adresse $1000. Sie werden nach dem Laden mit SYS 4096 aktiviert.
Der Joysticktreiber befindet sich im Kassetten-Puffer (ab 828/ hex $33c) und wird mit SYS 828 initialisiert.
Der Maus-Treiber liegt von 52887 bis 53247 ($cfff). Außerdem belegt er Sprite-Block 13 (832-896) für den Mauszeiger. Er wird mit SYS 52887 gestartet und beim erneuten Aufruf dieser Adresse deaktiviert.

Tabelle 1: Der Aufbau der Tastatur-Matrix

	56320/$dc00							
56321/$dc01	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bit 0	1	£	+	9	7	5	3	DEL
Bit 1	→	*	P	I	Z	R	W	RETURN
Bit 2	CTRL	;	L	J	G	D	A	CSR-L/R
Bit 3	2	CLR	-	0	8	6	4	F7
Bit 4	SPACE	SHIFT-R	.	M	B	C	Z	F1
Bit 5	CBM	=	:	K	H	F	S	F3
Bit 6	Q	I	KA	O	U	T	E	F5
Bit 7	RUN/ STOP	/	,	N	V	X	SHIFT-L	CSR-O/U

Tabelle 2: Die Bit-Belegung der Joystick-Ports 56320 und 56321

Bit	Richtung	Wert
0	Oben	1
1	Unten	2
2	Links	4
3	Rechts	8
4	Feuer	16

```
10 J%=0:IFA=O THEN A=1:THEN LO-
AD"JOY-DRIVER",8,1
20 SYS 828
```

Die Variablen J% und A sind ab jetzt Tabu und dürfen nicht mehr benutzt werden. Die Bedeutung der Werte von J% ermittelt man mit Tabelle 2. Das Grundgerüst für eine Joystick-Abfrage in Maschinensprache, finden Sie Listing 7. Eine Abfrgae mit Hilfe des ROR-Befehls in Maschinensprache wäre auch denkbar. Man rotiert den Akkumulator, prüft nur ob das Carry-Flag gelöscht ist und verzeigt. Eine elegante Methode, die aber durch die Rotate-Befehle einige Taktzyklen mehr verschlingt.

Die Maus-Abfrage

Die wesentlich höhere Geschwindigkeit bei der Positionierung des Mauszeigers, hat der Maus zum Durchbruch verholfen. Die Datenübertragung Maus/C 64 wird via Joystickport 1 (56321/hex. $dc01) realisiert. Die Koordinaten ermittelt man nicht über diese Register, sondern über die Speicherstellen 54297 (hex. $d419) und 54298 (hex. $d41a). Sie befinden sich im Soundchip (SID). Die Maus selbst simuliert veränderliche Widerstände, die der SID in Byte-Werte umwandelt. Die gemessenen Werte werden in den beiden genannten Speicherzellen abgelegt. Dabei verwendet der C 64 nur die sechs unteren Bits. Das reicht natürlich nicht für den Bildschirm des C 64, der bekanntlich für die Breite neun und für die Höhe acht Bits braucht.

Darum muß die Mausabfrage einige zusätzliche Berechungen ausführen, um die Distanzänderung des Mauszeigers zu ermitteln. Listing 8 zeigt einen kompletten Maus-Treiber, der im Interrupt läuft.

Die Mausbuttons werden über Joystick-Port 1 abgefragt. Wurde die linke Taste gedrückt, ist Bit 5 gelöscht, bei der rechten Bit 0. Damit alle Basic-Programmierer in ihre Programme ebenfalls eine Maussteuerung einbauen können, haben wir den Treiber unserer Diskette zum Heft beigegeben. Er positioniert den Mauszeiger im Interrupt und übergibt die Koordinaten und den Button-Status an drei Basic-Variablen.

Das Beispiel von Listing 9 zeigt, wie der Treiber nachgeladen und die Variablen definiert werden. **X%** enthält die horizontale Position des Zeigers, **Y%** die vertikale Position und **B%** den Button-Status (16= linker Knopf, 1=rechter Knopf, 0= kein Button).

Mit unseren Beispielen können Sie Ihre Programme sicher sogar noch ein wenig aufmöbeln und den Bedienungskomfort erhöhen. Vielleicht sind unserer Anregungen auch Grundstein für neue verbesserte Eingabe-Routinen.

Jörn Erik Burkert

Listing 1: Eine einfache Eingabe-Routine mit derGET-Funktion

```
20 PRINT"<SHIFT/CLR>"                  :REM BILDSCHIRM LOESCHEN
30 GETT$:IFT$=""THEN30                 :REM TASTE HOLEN
40 IF ASC(T$)<64ORASC(T$)>90THEN30     :REM KEIN ZEICHEN?
50 PRINTT$;:GOTO30                     :REM EINGABE AUF BILDSCHIRM
```

Listing 2: Listing 1 als Assembler-Version

```
key     jsr $ffe4       ;taste holen
        beq key         ;keine taste
        tax             ;sichern
        sec
        sbc #$40        ;kleiner "a"?
        bmi key         ;ja -> key
        sbc #$23        ;groesser "z"
        bpl key         ;ja -> key
out     txa
        jsr $e716
        jmp key
```


Dieses Symbol zeigt an, welche Programme auf Diskette erhältlich sind

Listing 3: Umwandlung des String-Inhalts in einen Zahlenwert

```
1000 REM STRING-ZAHL-WANDLUNG
1010 T$="12345":A=1:Z=0
1020 FOR I=1TO LEN(T$)
1030 : T=ASC(LEFT$(RIGHT$(T$,I),1))-48
1040 : Z=Z+T*A:A=A*10
1050 NEXT I
1060 PRINT T
```

Listing 4: Umwandlung eines Strings in Variable mit Nachkommastellen

```
1000 REM STRING-ZAHL-WANDLUNG MIT NACHKOMMASTELLEN
1010 T$="12.34":A=1:Z=0:K$="."
1020 REM KOMMA SUCHEN
1030 FOR I=1TO LEN(T$)
1040 IFK$<>LEFT$(RIGHT$(T$,I),1)THEN NEXT I
1050 K=I
1060 REM NACHKOMMASTELLE BERECHNEN
1070 FOR I=1TO K-1
1080 : T=ASC(LEFT$(RIGHT$(T$,I),1))-48
1090 : Z=Z+T*A:A=A/10
1100 NEXT I
1110 REM VORKOMMASTELLE BERECHNEN
1120 A=1
1130 FOR K+1 TO LEN$(T$)
1140 : T=ASC(LEFT$(RIGHT$(T$),I),1))-48
1150 : Z=Z+T*A:A=A*10
1160 NEXT I
1170 PRINT T
```

Listing 5: Abfrage des Keyboard mit Hilfe derTastatur-Matrix

```
          sei              ;irq sperren
          lda #<irq        ;irq-vektor
          sta $0314        ;verbiegen
          lda #>irq
          sta $0315
          cli              ;irq zulassen
          rts              ;zurueck
;-------------------------------------------------------------
irq       lda #%01111111   ;spalte
          sta $dc00        ;setzen
          lda $dc01        ;zeile lesen
          and #%00000100   ;ausmaskieren
          bne end          ;bit geloescht?
          inc $d020        ;key zeigen
end       jmp $ea31        ;zu irq-routine
```

Listing 6: Den Joystick in Basic im Griff

```
10 REM ABFRAGE JOYSTICK-PORT 2
20 J=INT(LOG(255.5-(PEEK(56320)OR224))LOG(2)+2)
30 ON J GOTO <zeilennummer>,<zeilennummer>...
```

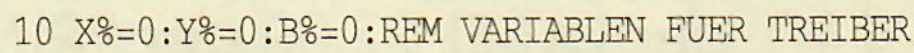
Listing 7: Die Joystick-Abfrage in Assembler

```
joystick  lda $dc00 ;port lesen
          and #%00011111   ;ausmaskieren
          eor #%00011111   ;invertieren
          beq joy          ;keine reaktion
          cmp #$01         ;oben?
          beq oben
          cmp #$02         ;unten?
          beq unten
          cmp #$03         ;rechts oben?
          beq roben
;alle weiteren richtungen und feuer analog anfuegen und
;abfrage vervollständigen
```



Listing 8: Mausabfrage in Assembler

```
start     lda #<spriteblock>
          lda #$01           ;maus-zeiger
          sta $d015          ;setzen
          lda #$a0
          sta $d000
          sta $d001
          sta xpos
          sta ypos
          sta oldpos
          sta newpos
          sei                ;irq-vektor
          lda #<irq          ;auf maus-abfrage
          sta $0314          richten
          lda #>irq
          sta $0315
          cli
          rts                ;zurueck
;-------------------------------------------------------------
irq       lda $d419          ;x-wert holen
          cmp xpos           ;alter wert?
          beq next           ;ja und weg
          ldy xpos           ;x-verschiebung
          jsr count          ;prüfen
          sty xpos           ;neue pos sichern
          clc
          adc $d000          ;zu alter addieren
          sta $d000          ;sprite setzen
          txa                ;hi-byte aus x holen
          adc #$00
          and #%00000001     ;ausmaskieren
          eor $d010          ;verknuepfen
          sta $d010          ;hi-byte schreiben
next      lda $d41a          ;y-wert holen
          cmp ypos                ;alter wert?
          beq end            ;ja und weg
          ldy ypos           ;y-verschiebung
          jsr count          ;pruefen
          sty ypos           ;neue pos sichern
          eor #%11111111     ;invertieren
          adc $d001          ;zu alten wert addieren
          sta $d001          ;sprite setzen
end       jmp $ea31          ;zu orig-irq
;-------------------------------------------------------------
count     ldx #$00
          sty oldpos         ;alten und neuen step
          sta newpos         ;speichern
          sec
          sbc oldpos         ;alten step abziehen
          and #%01111111     ;ausmaskieren
          cmp #%01000000     ;wenn bit gesetzt
          bcs cont1          ;überlauf prüfen
          lsr
          ldy newpos         ;neue position holen
          rts                ;zurueck
;-------------------------------------------------------------
cont1     ora #%11000000     ;ausmaskieren
          cmp #$ff           ;rechter rand erreicht
          bne cont2          ;nein und weiter
          lda #$00
          rts                ;zurueck
;-------------------------------------------------------------
cont2     sec
          ror
          ldx #$ff           ;hi-byte setzen
          ldy newpos         ;neue pos holen
          rts                ;zurueck
```

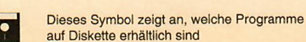


Listing 9: Die Einbindung des Maustreibers "MAUS.DRV" in eigene Basic-Programme

```
10 X%=0:Y%=0:B%=0:REM VARIABLEN FUER TREIBER
20 IF A=0THENA=1:LOAD"MAUS.DRV",8,1
30 SYS SYS52887:REM START MAUSTREIBER
40 REM EIGENES BASIC-PROGRAMM STARTET HIER
```






Dieses Symbol zeigt an, welche Programme auf Diskette erhältlich sind

Grundlagen

Die Sache mit dem IRQ

Spiele oder Demos nutzen in den meisten Fällen einen Effekt, der auf der Positionsabfrage des Kathodenstrahls beruht. Damit lassen sich z.B. "Rastersplits" programmierern, die für gleichzeitige Bildschirmdarstellung von Text- und HiRes-Grafik sorgen.

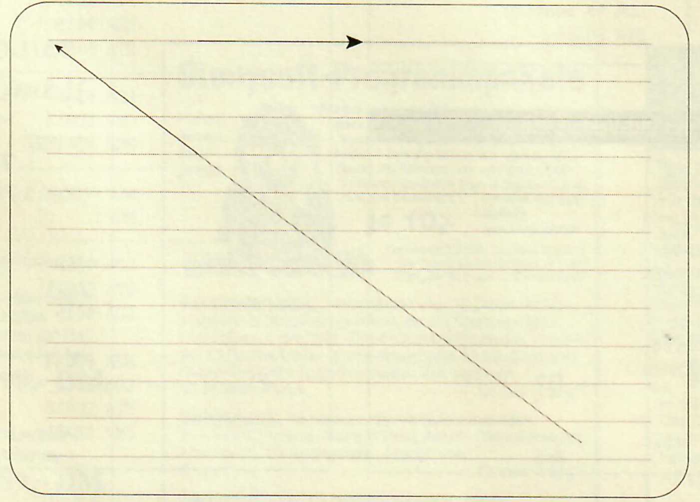
Um die Funktionsweise eines Interrupts zu verstehen, ist die Kenntnis einiger Grundlagen unumgänglich. Wie Sie wahrscheinlich wissen, generiert der C64 mit Hilfe des Grafikchips VIC ein Bild und schickt es dann an den Monitor bzw. Fernseher. Der erzeugt das eigentliche Bild mit Hilfe einer chemisch vorbehandelten Leuchtschicht und einem Kathodenstrahl: dieser Strahl (auch "Rasterstrahl" oder "Elektronenstrahl" genannt) besteht aus einer Unmenge Elektronen, die mit nahezu Lichtgeschwindigkeit auf die erwähnte Leuchtschicht treffen. Eine Ablenkungseinheit sorgt nun dafür, daß die Elektronen zeilenweise und von oben nach unten korrekt an der dafür vorgesehenen Stelle ankommen. Ist der Elektronenstrahl am unteren Bildrand angekommen, wird er wieder in die linke obere Ecke plaziert und das ganze Spiel beginnt von vorne. Der VIC ist dabei immer bestens informiert, an welcher Position der Kathodenstrahl gerade seines Amtes waltet. Diese Information vermittelt er auch dem Programmierer, indem er die Y-Koordinate (entspricht der Zeile) im Register $D012 (dez. 53266) speichert. Für die exakte X-Position (entspricht der Spalte) ist der Grafikchip allerdings zu langsam - immerhin werden bereits für den Aufbau einer kompletten Zeile nur 64 Millisekunden verbraten!

Was liegt nun näher, als einfach dieses Register stur abzufragen und bei einer bestimmten Position zu einem beliebigen Unterprogramm zu verzweigen? In Listing 1 finden Sie die entsprechende Routine. Bevor Sie jetzt allerdings zu Ihrem Computer spurten, um das Gelernte in die Tat umzusetzen, sollten Sie sich zuerst mit dem Interrupt des C 64 vertraut machen.

Der IRQ

Wenn Sie Ihren Rechner anschalten, erscheint nach kurzer Zeit die vertraute READY-Meldung und der Cursor blinkt. Betätigen Sie eine Taste, wird das jeweilige Zeichen ausgegeben. Damit der Computer jederzeit auf diverse Eingaben reagieren kann, unterbricht er seine Arbeit alle 1/60stel Sekunden. Dieser Vorgang wird Interrupt oder IRQ genannt. Er läßt den Cursor blinken, wartet auf eine Tastatureingabe usw.

Die Routine, die diese Aufgabe übernimmt, steht ab $EA31 im Kernel (C-64-Betriebssystem).



Der Rasterstrahl baut den Bildschirm zeilenweise auf und begibt am Screenende wieder nach links oben

Nach einem Interrupt wird jedoch nicht immer automatisch an diese Adresse verzweigt! Das Betriebssystem holt sich aus den Adressen $0314 und $0315 (dez. 788 und 789) das Low- und Highbyte der Sprungadresse (Vektor-Adresse). Wollen Sie also beispielsweise eine eigene Routine nach einem Interrupt ausführen, müssen Sie lediglich dafür sorgen, daß in den genannten Adressen die Anfangsadresse Ihres eigenen Unterprogramms steht. Listing 2 zeigt diesen Vorgang.

Die "<"- bzw ">"-Zeichen bedeuten in diesem Fall Low- und Highbyte. Aber Achtung: Wenn Sie diese Befehlsfolge mit einem SYS ausführen lassen, kann es durchaus vorkommen, daß der Computer ausgerechnet in dem Moment einen Interrupt ausführt, wenn gerade das Lowbyte geschrieben wird. Da nun eine unvollständige und damit unrichtige Adresse in den Vektoren steht, gibt's einen sauberen Absturz zu bewundern. Wir müssen also sicherstellen, daß der C 64 keinen IRQ während der Abarbeitung der vier Befehlszeilen auslöst. Mit den Assemblerbefehlen *SEI* sperren wir den normalen IRQ, mit *CLI* geben wir ihn wieder frei. Im (Klar-)Quelltext sieht das wie in Listing 3 aus.

Jetzt müssen wir dem VIC nur noch mitteilen, daß wir als Interrupt-Quelle lediglich einen Raster-IRQ akzeptieren und unser erstes IRQ-Programm ist wie in Listing 6 zu sehen komplett.

Die Original-Einsprungsadresse $EA31 rufen wir auf, damit weiterhin Tastatureingaben möglich sind und der Cursor scheinbar gleichzeitig zu unserem Farbensplit blinkt. Wenn Sie das Programm ausführen, hat sich der Bildschirm in zwei farbige Bereiche aufgeteilt.

Achten Sie übrigens darauf, für Rasterstrahl-Positionen immer nur Hexzahlen anzugeben, die mit 0 oder 8 enden (z.B. $F0, $30, $C8 usw.). Damit vermeiden Sie unschönes Geflacker.

Der Text-Grafiksplit

Ob es sich bei einem solchen Split um Farben oder beispielsweise um verschiedene Grafikmodi handelt, ist egal. Wichtig ist nur, daß Sie die korrekte Rasterzeile entweder durch Ausprobieren oder mit einem Schaubild ermitteln und dann entsprechend in Ihre Abfrage einbauen. Wollen Sie ein Grafikadventure mit Texteinlagen programmieren, ist eine Aufteilung des Bildschirms in Text- bzw. Grafikbereich ratsam. Sie ersparen sich damit eine umständliche Routine zum Plotten der Buchstaben in den Grafikbildschirm. Listing 5 zeigt, wie die Teilung des Bildschirms funktioniert.

Die Initialisierungsroutine von unserem vorherigen Beispiel bleibt identisch. Sie benötigen jedoch noch eine kleine Routine, die das Bild (in diesem Fall im Koala-Format) an die korrekten Adressen setzt. Ein Multicolorbild setzt sich immer aus drei verschiedenen Blöcken zusammen: der eigentlichen Bitmap, dem Farb- und Bildschirm-RAM. Die Bitmap liegt, wenn Sie das File ab $2000 in den Speicher holen, bereits an der korrekten Adresse (ab $2000), der Bereich von $3F40 bis $4327 muß ins Screen-RAM (ab $0400) kopiert werden und die Bytes von $4328 bis $470F gehören ins Color-RAM (ab $D800).

Mit unserem Trick schalten wir nun ab Rasterzeile $38 (entspricht dem oberen Bildrand) unsere Grafik ein: in Adresse $D011 wird mit Bit 5 der Grafikmodus aktiviert, mit $D016 schalten wir auf Multicolor-Modus um und in $D018 geben wir dem VIC an, wo er die eigentliche Grafikbitmap zu suchen hat. Ab Rasterzeile $A2 machen wir dann alle Aktionen wieder rückgängig.

Nach Start des Programms wird in den oberen zwei Dritteln des Screens das im Speicher befindliche Multicolor-Bild eingeblendet, das untere Drittel steht für Texteingaben zur Verfügung. Ach ja: die beiden NOPs sind als Rasterzyklenausgleich dringend notwendig um ein evtl. Flackern zu vermeiden. Das exakte Timingverhalten ist je nach C-64-Modell

übrigens hauchfein unterschiedlich, deshalb kann es trotz dieses Ausgleichs bei manchen Geräten flackern. Hier heißt es: Probieren geht über studieren!

Die Rasterbars

Etwas komplizierter als einfache Splits sind "Copper-" oder "Rasterbars". Wer nämlich nach obigem Schema versucht, die berühmten feinabgestuften Farbbalken auf seinen Monitor zu bringen, wird nach kurzer Zeit entweder verrückt oder wirft das Handtuch. Warum aber flackern trotz korrekter Abfragen ganze Bereiche? Grund allen Übels: wieder einmal - der VIC! Wir haben gelernt, daß eine Zeile normalerweise in 64 Millisekunden (entspricht 64 Taktzyklen) aufgebaut wird. Alle acht Zeilen (acht Zeilen ergeben die Höhe eines Buchstabens) liest der VIC jedoch das Screen- und Color-RAM aus und schickt die gewonnenen Informationen auf die Reise an den Bildschirm. Da der zusätzliche Vorgang mehr Zeit als normal beansprucht, kommt das komplette Timing durcheinander. Wir brauchen also eine "dynamische" Warteschleife, die die unterschiedlichen Zeiten wieder ausgleicht. Diesen Operation nennt man übrigens "Rasterzyklenausgleich". Eine solche Warteschleife sehen Sie in Listing 4.

Unsere Routine ist fertig und zaubert einen goldenen Bereich an den oberen Rand des Bildschirms. Übrigens: Wenn Sie den Balken wandern lassen wollen, müssen Sie lediglich dafür sorgen, daß die Farbtabelle (nicht die Wartetabelle) im Speicher rotiert wird. Wenn Sie die Tabelle nach links schieben (achten Sie darauf das erste Byte zu retten und am Tabellenende wieder anzusetzen), wandert der Balken nach unten. Eine Verschiebung der Tabelle in die andere Richtung, sorgt für eine Bewegung des Balken nach oben. Mit Hilfe eines Copper-Bar-Editor lassen sich Farbtabellen kinderleicht erstellen und später in eigene Programme einbauen. Solche Hilfsprogramme finden Sie in jeder guten Public-Domian-Sammlung.

Natürlich können Sie auch andere Routinen in den Interrupt hängen und so z.B. die "Kontrolle" über den C 64 an sich ziehen.

Peter Klein/lb

Listing 1: Abfrage des Rasterstrahls

```
        LDA #$40
WEITER  CMP $D012
        BNE WEITER
        JSR Unterroutine
```

Listing 2: Verbiegen des IRQ-Sprungvektors

```
LDA #< Routine
STA $0314
LDA #> Routine
STA $0315
```

Listing 3: Die Initialisierung des IRQ-Vektors komplett

```
SEI             ;IRQ verhindern
LDA #<SPLIT     ;Lowbyte der
                ;Adresse holen
STA $0314       ;und in Lowbyte-
                ;Vektor speichern
LDA #>SPLIT     ;Highbyte der
                ;Adresse holen
STA $0315       ;und in Highbyte-
                ;Vektor speichern
CLI             ;IRQ wieder
                ; freigeben
RTS             ;zurück ins Basic
```

Listing 5: Teilung des Bildschirms für Grafik und Text

```
SPLIT   LDA #$38
AGAIN   CMP $D012
        BNE AGAIN
;----------------------------------------------------
        LDA #$1D        ;Bitmap-Bereich
        STA $D018       ;setzen
        LDA #$3B        ;Bitmap
        STA $D011       ;aktivieren
        LDA #$18        ;Multicolor-Modus
        STA $D016       ;anschalten
;----------------------------------------------------
        LDA #$A2
AGAIN2  CMP $D012
        BNE AGAIN2
;----------------------------------------------------
        NOP             ;Rasterzyklenausgleich
        NOP             ;verhindert Flackern
;----------------------------------------------------
        LDA #$1B        ;Bitmap-Modus
        STA $D011       ;deaktivieren
        LDA #$15        ;Original-
                        ;Zeichensatz
        STA $D018       ;einschalten
        LDA #$C8        ;Multicolor-Modus
        STA $D016       ;ausschalten
        JMP $EA31
```

Listing 4: Goldene Farbbalken per Rasterbar-Effekt

```
        SEI             ;IRQ verhindern
        LDA #<BAR       ;Lowbyte der
                        ;Adresse holen
        STA $0314       ;und in Lowbyte-
                        ;Vektor speichern
        LDA #>BAR       ;Highbyte der
                        ;Adresse holen
        STA $0315       ;und in Highbyte-
                        ;Vektor speichern
        LDA #$81        ;nur Raster-IRQ
        STA $D01A       ;zulassen
        CLI             ;IRQ freigeben
        RTS             ;zurück ins Basic
;----------------------------------------------
BAR     LDA #$40        ;Position $00 laden
AGAIN   CMP $D012       ;und mit der
                        ;aktuellen
                        ;Rasterstrahl-
        BNE AGAIN       ;Position
                        ;vergleichen
;----------------------------------------------
        NOP             ;Rasterzyklenausgleich
;----------------------------------------------
        LDX #$00
WAIT1   LDY WAITAB,X    ;Element aus Warte-
                        ;Tabelle holen
WAIT2   DEY             ;herunterzählen,
        BNE WAIT2       ;auf Wert $00
        LDA COLTAB,X    ;Element aus
                        ;Farbtabelle holen
        STA $D020       ;und in den
                        ;Bildschirm bzw.
        STA $D021       ;Hintergrund
                        ;schreiben
        INX
        CPX #$0F        ;schon alle
                        ;Elemente?
        BNE WAIT1       ;nein, dann wieder
                        ;bei WAIT1 weiter
;----------------------------------------------
        LDA #$00        ;Farben auf
        STA $D020       ;Schwarz
        STA $D021       ;setzen
;----------------------------------------------
        JMP $EA31       ;zu Original-
                        ;Kernel-Routine
;----------------------------------------------
COLTAB  .BYTE $00,$09,$0B,$08,$0C,$0F,$07,$01
        .BYTE $01,$01,$07,$0F,$0C,$08,$0B,$09
;----------------------------------------------
WAITTAB .BYTE $09,$08,$08,$01,$08,$08,$08,$08
        .BYTE $08,$08,$08,$01,$08,$08,$08,$08
```


Dieses Symbol zeigt an, welche Programme auf Diskette erhältlich sind

Listing 6: IRQ-Routine zur Umschaltung der Bildschirmfarben

```
         SEI
         LDA #<SPLIT
         STA $0314
         LDA #>SPLIT
         STA $0315
         LDA #$81         ;nur Raster-IRQ
                          ;zulassen
         STA $D01A        ;Interrupt Mask
                          ;Register
         CLI
         RTS              ;zurueck zu Basic
;-----------------------------------------------
SPLIT    LDA #$00         ;Position $00 laden
AGAIN    CMP $D012        ;und mit der
                          ;aktuellen
                          ;Rasterstrahl-
         BNE AGAIN        ;Position vergeleichen
;-----------------------------------------------
         LDA #$02         ;Farbe $02
         STA $D020        ;in Bildschirm und
         STA $D021        ;Bildschirmrand
                          ; schreiben
;-----------------------------------------------
         LDA #$A0         ;Position $A0 laden
AGAIN2   CMP $D012        ;und vergleichen
         BNE AGAIN2
;-----------------------------------------------
         LDA #$03         ;Farbe $03
         STA $D020        ;in Bildschirm und
         STA $D021        ;Bildschirmrand
                          ;schreiben
         JMP $EA31        ;zu Original-Kernel
                          ;springen
```

© 64'er







Dieses Symbol zeigt an, welche Programme auf Diskette erhältlich sind



Synthetische Steuerzeichen" sorgten vor Jahren für viel Wirbel in der Programmierszene um den c 64. Mit ihnen können Sie Listschutz-Effekte in Basic kinderleicht generieren oder Ihre Listings optisch aufpeppen.

Die Steuerzeichen des C 64 sorgen in Strings z.B. für Farbveränderungen oder die Umschaltung in den Kleinschrift-Modus. Das bekannteste Steuerzeichen dürfte das reverse Herz (Tastenkomb. SHIFT+CLR) sein: Es löscht den Bildschirm. Dieses Zeichen läßt sich in einem String generieren, wenn er mit einem " geöffnet wurde. Jetzt reagieren z.B. die Cursor-Tasten nicht mehr wie gewohnt, sondern erzeugen seltsame reverse Grafik-Zeichen. Beim Abarbeiten interpretiert der C 64 die Zeichen als Cursor-Bewegungen. Eine Positionierung von Text auf dem Bildschirm ist so ein Kinderspiel. Was aber geschieht, wenn man andere reverse Zeichen im String integriert?

Das ist recht schwierig, denn beim Tippen erscheinen die Buchstaben und Zahlen wie gewohnt. Um andere Steuerzeichen zu erzeugen "schließen" wir die String-Eingabe mit einem " ab und setzen die Eingabe mit der DEL-Taste zurück. Nun begeben wir uns mit der Tastenkombination <CTRL+9> in den Revers-Mode und können nach Herzenslust invertierte Zeichen eingeben. Nach der Eingabe dieser "synthetisch" erzeugten Steuerzeichen, deaktiviert man den Revers-Modus mit der Tasten-Kombination <CTRL+0> wieder und übergibt mit der RETURN-Taste die Basic-Zeile an den Speicher.

Das reverse "T"

Mit dem reversen "T" wird in Strings die DEL-Taste simuliert und kann zum Verstecken von Befehlen in Listings dienen. Tippen Sie einfach im Eingabemodus folgendes ein und bestätigen Sie die generierte Basic-Zeile mit der RETURN-Zeile:

```
10 POKE53280,0: REM"" <DEL>
<CTRL+9> <16xT>
```

Nun können Sie das kleine Programm mit dem RUN-Befehl starten und werden sehen, daß der Bildschirm-Rahmen schwarz wird. Als nächstes wollen wir versuchen, den Farbwert im POKE-Befehl zu ändern. Einfach LIST eingeben und den Wert hinter dem Komma überschreiben...

Eine Modifizierung ist aber unmöglich, da der Befehl schlicht verschwunden ist. Was ist passiert? Die reversen "T"s in der REM-Zeile machen die Ausgaben beim LISTen wieder rückgängig. Dieser Effekt funktioniert, da das reverse "T" ja für die DEL-Taste steht. Man kommt also nicht mehr an den POKE-Befehl heran. Eigentlich ein perfekter LIST-Schutz!

Der LIST-Schutz

Die Methode mit dem reversen "T" ist bei sehr langen Listings auf Dauer sehr arbeitsintensiv. Außerdem verschwendet man wertvollen Speicherplatz im Basic-Bereich.

Programmierer, die sich aber trotzdem nicht in die Karten schauen lassen wollen, sollten an die zu schützenden Basic-Zeilen

```
REM""<DEL><CTRL+9><SHIFT+M>
<SHIFT+S><CTRL+0>
```

anfügen.

Beim LISTen der Basic-Zeilen mit dem Anhängsel, wird nun der komplette Bildschirm gelöscht und die Zeilen werden so für den Betrachter versteckt.

Die beiden vorgestellten Tricks dürften beim LISTen für einige Verwirrung sorgen. Für User mit Erfahrung in der Arbeit mit Maschinensprache-Monitoren jedoch, wird dieser Schutz kein großes Hindernis sein. Mit einem Monitor werden blitzschnell die Bytes für den LIST-Schutz entfernt und das Listing wird wieder sichbar.

```
10 POKE53280,0:REM"▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮
RUN

READY.
LIST

10
READY.
SYS49152

  PC  SR AC XR YR SP  NV-BDIZC
;C00B B0 C2 00 00 F6  10110000
.M 0800 0830
:0800 00 22 08 0A 00 97 35 33   .".. .:53
:0808 32 38 30 2C 30 3A 8F 22   280,0:."
:0810 14 14 14 14 14 14 14 14   ........
:0818 14 14 14 14 14 14 14 14   ........
:0820 14 00 00 00 00 00 00 00   ........
:0828 00 00 00 00 00 00 00 00   ........
.
```

Mit einem Maschinensprache-Monitor lassen sich die synthetischen Steuerzeichen (hier $14) schnell finden und bei Bedarf entfernen

Variablen im Interrupt übergeben

Die Übergabe von Integer-Variablen im Interrupt läßt sich nicht über die Basicfunktion LET realisieren. Der C 64 stürzt ab. Folgende Routine sorgt für den gewünschten Effekt:

```
start   sei        ;irq-init
        lda #<irq
        sta $314
        lda #>irq
        sta $315
        cli
        rts
irq     lda#"J"    ;J% init
                   ;unbedingt
                   ;gr. "J"
        jsr fvar   ;variable
                   ;suchen
        lda wert
        sta ($fb),y
        dey        ;J% zuweisen
        lda wert+1
        sta ($fb),y
        jmp $ea31
wert    .byte 0,0
fvar    sta such+1
        lda $2d
        sta $fb
        lda $2e
        sta $bc

floop   ldy #0
        lda ($fb),y
such    cmp #$ff
        bne fcont
        iny
        lda ($fb),y
        cmp #$80  ;%-Zeichen?
        beq found
fcont   lda $fb
        clc
        adc #$07
        sta $fb
        bcc find
        inc $fc
find    cmp $2f
        bne floop
        lda $fc
        cmp $30
        bne floop
nofound rts
found   iny
        iny
        rts
```

Die Variable J% muß in Basic gesetzt werden und erst dann kann die Initialisierung des Maschinenprogramms erfolgen. Natürlich können Sie auch J% mit Hilfe eines Maschinensprache-Programms setzen. *Maxim Szenessy*

Effektvolles LISTen

Für Programmierer ist gute Übersicht beim LISTen eines Programms schon die halbe Miete. Einige synthetische Steuerzeichen sorgen beim LISTen für interessante Effekte:

```
10 REM"" <DEL> <CTRL+9>
<SHIFT+M> <SHIFT+Q> <R>
<CTRL+0> TEXT
```

Wenn sie nach der Eingabe der Basic-Zeile, das Programm LISTen, wird die Zeilennummer und die REM-Anweisung "verschluckt". Wie funktioniert das? Ganz einfach: das Steuerzeichen <SHIFT+M> setzt den Cursor (nach der Ausgabe der REM-Anweisung) an den Anfang der nächsten Zeile.

Das nun folgende Steuerzeichen (<SHIFT+Q>) befördert den Cursor wieder eine Zeile nach oben. Nun folgt das Steuerzeichen für reverse Bildschirmdarstellung und der restliche Text im String wird ausgegeben. Abschließend wird der Revers-Mode beendet.

Wer die Ausgabe farblich noch ein wenig gestalten will, gibt dem String nach der SHIFT+Q-Anweisung noch eine Farbanweisung mit. Dazu fügt man nur das entsprechende Steuerzeichen ein:

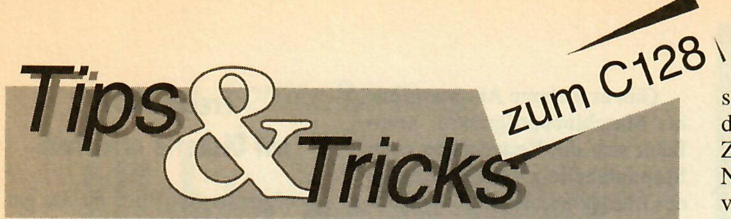
```
10 REM"" <DEL> <CTRL+9>
<SHIFT+M> <SHIFT+Q> <E><R>
<CTRL+0>TEXT<CTRL+9>< SHIFT+Z>
<CTRL+0>
```

Beim LISTen wird der Kommentar in Weiß ausgegeben und der folgende Text in Grau. Die entsprechenden Steuercodes für die Farben finden Sie im C-64-Handbuch.Die vorgestellten Steuerzeichen lassen sich natürlich kombinieren und führen so zu weiteren Effekten.

Vielleicht tüfteln Sie ein wenig, schreiben uns und finden demnächst Ihren Steuerzeichen-Tip an dieser Stelle.

Jörn-Erik Burkert





Heute stellen wir einen Zeichensatz-Editor im 40-Zeichen-Modus vor, mit dem sich komfortabel neue Charsets entwerfen lassen. Der Clou: diese geänderten Zeichensätze kann man mit entsprechendem Ladeprogramm auch im 80-Zeichen-Bildschirm einsetzen!

Ed'Char 128

Der Zeichensatz-Editor arbeitet ausschließlich im 40-Zeichen-Modus und besteht aus folgenden Programmteilen:

- ❑ EDCHAR.128 (Hauptprogramm)
- ❑ CHAR.SPR (Sprites für Editor-Cursor und Titelbild)
- ❑ ZS.USER (Assembler-Routinen zum Hauptprogramm)
- ❑ AMIGA.ZS (Demo-Charset).

Vor dem Starten mit:

```
RUN "EDCHAR.128"
```

sollten Sie den Resetknopf drücken. Der Basic-Anfang muß unbedingt bei $1C01 (7169) liegen, sonst gibt's Probleme beim Programmablauf.

Vor dem eigentlichen Editieren müssen Sie entscheiden, ob Sie ASCII- oder DIN-Zeichenmuster bearbeiten wollen (Taste A oder D). Die DIN-Taste muß nicht extra gedrückt werden! Anschließend können Sie den Original-Zeichensatz oder einen geänderten von Diskette ins RAM ab $3000 (12288) laden – denn nur dort lassen sich die Muster ändern.

Das gewünschte Zeichen gibt man über die Tastatur ein - unmittelbar darauf steht der Editierbildschirm zur Verfügung (64fache Vergrößerung des Originalzeichenmusters).

Die Zahlen unter dem Matrixfeld kennzeichnen die jeweiligen Bit-Positionen, aus denen die entsprechenden Dualwerte berechnet werden. Ganz oben links wartet ein hellblaues kreuzförmiges Sprite, daß es mit den jeweiligen Cursor-Tasten innerhalb des Entwurffelds in alle vier Hauptrichtungen bewegt wird.

Trägt man per Sternchen-Taste ein Bit ein, verschwindet der Rasterpunkt und das Bit erscheint als weißes Feld (beim Löschen geht's genau umgekehrt).

Beachten Sie: alle Änderungen innerhalb des Editorfelds werden erst dann im Computerspeicher eingetragen, wenn Sie die gesamte aktuelle Bitreihe (also ein Byte) per <RETURN> übernehmen. Als aktuelle Bitreihe gilt immer die, in der sich der Editor-Cursor gerade aufhält.

Wer wissen will, wie das geänderte Zeichen in Originalgröße aussieht, sollte einen Blick auf die Bildschirmmitte werfen: dort sind alle Zahlen, Buchstaben und Grafikzeichen (Bildschirmcodes 0 bis 127) hintereinander abgebildet. Auf die Code-Werte 128 bis 255 haben wir nicht nur aus Platzmangel auf dem Arbeitsbildschirm verzichtet: diese Zeichen bieten nicht anderes als die reverse Anzeige der Zeichen Nr. 0 bis 127. Und deren Änderung übernimmt automatisch die Maschinensprache-Routine ZS.USER.

Jeder geänderte und auf Disk gesicherte Zeichensatz läßt sich problemlos wieder laden:

```
BLOAD "(File-Name)",ONB0,
P(gewünschte Ladeadresse, sonst
$3000)
```

Hinweise zum Programm

Ab sofort kann man das neue Charset in eigenen Programmen verwenden. Wichtig ist, daß Sie dem C 128 über die Systemadresse $0A2C (2604) Start und Lage des neuen Zeichensatzes mitteilen. Zuständig dafür ist das untere Nibble (Bits 0 bis 3) des Inhalts von $0A2C. Der entsprechende Wert errechnet sich aus:

Startadresse/1024

Liegt der Zeichensatz also bei $3000 (12288), ergibt sich der Wert "12". Diese Zahl muß jetzt im Low-Nibble dieses Bytes stehen, die oberen vier Bits (4 bis 7) werden mit der AND-Verknüpfung maskiert:

```
POKE 2604,(PEEK(2604)
AND240)OR 12
```

Wer auf die Basic-Anweisung verzichten will, kann die Speicherstelle $0A2C selbstverständlich auch im Tedmon durch überschreiben der Hex-Bytes ändern.

Als Beispiel finden Sie einen geänderten Zeichensatz auf Diskette: AMIGA.ZS, der nach den Mustern des Amiga-Zeichensatzes (Schrifttyp TOPAZ) entworfen wurde.

Drückt man die Tastenkombination <RUN/STOP RESTORE>, werden die veränderten Interrupt-Vektoren wieder geradegerückt (und Adresse $0A2C erhält wieder den Originalwert - die Zeichenmuster kommen jetzt wieder aus dem Bereich ab $D000 in Bank 14). Man muß also die geänderten Zeichenmuster erneut durch Manipulation der Speicherzelle $0A2C aktivieren.

Ändern Sie nichts am Basic-Teil des Hauptprogramms "Ed'Char 128": die Maschinensprache-Routine enthält Sprünge auf diverse Zeilennummern im Basic-Programm (man sollte diese Zahlen also nicht ändern!). Das ermöglicht eine Betriebssystem-Routine im ROM des Basic-Interpreters: GOTO ab $59DB (Bank 15), die allerdings ein bißchen modifiziert werden mußte.

Ed'Char 128 (Editor-Funktionen)

Taste	Funktion
Cursor-Tasten	gewünschtes Bit im vergrößerten 8x8-Pixel-Feld an wählen
<*>	Bit setzen (einschalten)
<SPACE>	Bit löschen
<SHIFT CLR/HOME>	alle aktiverten Bits im Editierfeld löschen
<RETURN/ENTER>	aktuelle Bitreihe in den Speicher
<O>	Original-Zeichensatz vom ROM ins RAM (ab $3000) kopieren
<Z>	Zeichenmuster im RAM ab $3000 bearbeiten
<A>	geänderten Zeichensatz von Disk laden
<N>	anderes Zeichen im Editorfeld zeigen
<S>	aktuelle Charsets, die vorher nach $3000 bis $3FFF kopiert (und evtl. verändert wurden), speichert man nach Eingabe eines markanten File-Namens auf Disk. Als Lade- bzw. Startadresse der Zeichensatz-
Datei wird	stets $3000 eingetragen. Beachten Sie: Auch, wenn man nur einige Bytes geändert hat, wird immer der ge samte Zeichensatz (4096 Bytes) gesichert – man braucht also stets mindestens 17 freie Blocks auf Disk!
<E>	Programmende. Die Pointer des Betriebssystems zei gen wieder auf den Original-Zeichensatz im ROM ab $D000 (Bank 14).

Neue Zeichensätze fürs VDC-RAM

Obwohl unser Charakter-Editor nur im 40-Zeichenmodus läuft, läßt sich damit jeder neu entworfene bzw. geänderte Charset auch ins VDC-RAM übertragen und auf dem 80-Zeichen-Screen zeigen. Hierbei ist jedoch zu beachten, daß beim Laden und Speichern stets das Prinzip der sequentiellen Datenübertragung zu beachten ist: man legt den Startbereich des Daten-Transfers fest, öffnet den Datenkanal und überträgt die vorgesehene Byte-Anzahl. Allerdings lassen sich die im C-128-Betriebssystem implementierten Routinen $FFD5 (Laden) und $FFD8 (Speichern) im Zusammenhang mit dem VDC-RAM nicht nutzen – dazu muß man selbst zur Tastatur greifen und ein Programm entwerfen (wie beispielsweise unser Utility auf der Programmservice-Diskette, daß allerdings nur im 80-Zeichenmodus funktioniert):

```
BLOAD "ZS-LADER.VDC"
```

Es belegt den Bereich von $1300 bis $13D2 im Computer. Man aktiviert das Utility mit:

SYS 4864.

Nach Eingabe des Dateinamens des geänderten Zeichensatzes holt der Computer die Daten Byte für Byte von Diskette und überschreibt das Original-RAM des VDC ab $2000.

Tauchen die READY-Meldung und der Cursor wieder auf dem Screen auf, ist der neue Charset geladen. Ab sofort erscheinen alle Zeichen auf dem Bildschirm in neuem Gewand – daran ändert nicht einmal die Tastenkombination <RUN/STOP RESTORE> etwas (im Gegensatz zum VIC-Chip). Nur vor einer Taste sollten Sie sich hüten: <ASCII/DIN>. Ein Tipp darauf aktiviert den DIN-Zeichensatz und löscht den neuen – beim Entriegeln wird wieder der Original-Charset aus dem ROM geholt. Dann bleibt nur noch, den geänderten Zeichensatz per ZS-LADER.VDC erneut ins VDC-RAM zu holen und zu aktivieren.

Denken Sie daran: Zeichensatz-Änderungen beschränken sich nicht nur auf den Entwurf anders gestalteter Buchstaben oder Zahlen – man kann mit geänderten Zeichensätzen auch ganze Spiele-Landschaften zusammenstellen, wenn man bestimmte, selten benutzte Grafikzeichen z.B. als Mauerblöcke, Grasbüschel, UFOs, Aliens definiert und separaten Zeichensatz auf Disk verewigt.

Harald Beiler

Heiße Tips für den Plus/4

Trick-Parade

Der Plus/4 ist noch lange nicht ausgereizt: das beweist unsere raffinierte Tricksammlung in dieser Ausgabe. Ausgewählt wurde sie von Christian Schäffner.

Directory-Auswahl

Oft muß man bei Basic-Programmen einen File-namen angeben, um z.B. ein Bild zu laden oder eine Datei zu öffnen. Aber - wie wurde er geschrieben? Ist das Programm überhaupt auf Diskette? Also Programm abbrechen und nachsehen!

Einfacher geht es, wenn man den Programmnamen direkt aus dem Directory auf dem Bildschirm übernimmt. Das ist möglich, wenn man den Basic-Editor benutzt, der durch den INPUT-Befehl repräsentiert wird. Dazu muß man das Fragezeichen abschalten (das erledigt die Anweisung POKE 19,5). Die TRAP-Fehlerabfrage ist notwendig, um bei längeren Directories die Liste an passender Stelle mit <RUN/STOP> ohne Fehlermeldung abzubrechen:

Beim DIRECTORY-Befehl kann man durch Namensvorgabe und Verwendung der Joker ohne weiteres nur bestimmte Files anzeigen. So ermöglicht

```
DIRECTORY"T*=S"
```

die Anzeige aller sequentiellen Files, die mit "T" beginnen.

Wie beim normalen Directory-Befehl läßt sich auch hier die Ausgabe mit <CTRL S> anhalten oder per Commodore-Taste erheblich verlangsamen.

Basic- und Maschinensprache

Viele Basic-Programme brauchen Unterstützung durch Maschinenprogramme, um zeitkritische Routinen optimal durchzuführen. Doch wie bringt man Basic- und Maschinensprache-Teil am besten zusammen? Das Nachladen des MC-Programms ist zwischenzeitlich "out", da man erhebliche Probleme bekommt, wenn das Programm in EPROM- oder sRAM-Erweiterung gebrannt werden (hier ist Nachladen unmöglich). Am einfachsten ist, das Maschinenprogramm direkt ans Basic-Programm zu hängen und das Ende der Basic-Datei so zu verschieben, daß die numerischen Variablen erst hinter dem MC-Teil angelegt werden. Das funktioniert aber nur, wenn sich beide Programmteile im Speicher unter $8000 befinden.

Das Maschinenprogramm sollte man unbedingt per Assembler entwerfen (Hypra Ass, Turbo Ass, SVS Macro Assembler), da man dessen Startadresse eventuell noch verschieben muß, falls das Basic-Programm länger wird. Es empfiehlt sich, zwischen Basic und MC noch 256 bis 512 Byte frei zu lassen, damit kleinere Änderungen am Basic-Programm noch problemlos möglich sind, ohne den Maschinencode jedesmal neu assemblieren zu müssen. Mindestens aber drei Nullbytes sollten nach der letzten Basic-Zeile eingetragen werden. Das aktuelle Ende des Basic-Programms steht in relevanten Speicherstellen $2D/$2E, die das Betriebssystem verwaltet.

Gibt es mehrere Ansprungziele im Maschinenprogramm, empfiehlt sich die Verwendung einer Sprungtabelle gleich zu Beginn des Maschinencodes. So läßt sich in Basic die Startadresse in einer Variablen festlegen, zu der bei jedem SYS-Aufruf nur ein fixierter Wert zu addieren ist. Verschiebt sich irgendwann einmal der Maschinencode, muß nur die Adresse bei der Basic-Variablenzuweisung geändert werden, aber noch lange nicht jeder SYS-Befehl. Bei Verschiebungen im Maschinencode selbst muß man dann im Basic-Teil gar nichts ändern.

Dazu ein Beispiel:

```
10 S=8192
...
1010 SYS S
...
1999 SYS S+3
```

Laden: Das Basic-Programm wird normal geladen. Um den Maschinencode dranzuhängen, wird dieser einfach mit LOAD"name",8,1 geladen. Dabei werden die relevanten Zeiger automatisch gesetzt. Achtung: Der L-Befehl im Monitor macht das nicht!

Wenn man nur eine kleine Änderungen des Basic-Codes durchführen will, muß man den Maschinensprache-Teil vorher wieder abkoppeln – sonst wird er mitverschoben. Das erledigt der OLD-Befehl (OS92.x) oder die Basic-Anweisung:

```
POKE4098,1:DEL1
```

Jetzt ist das Programmende in den Speicherstellen $2D/$2E per POKE-Anweisungen wieder aufs Maschinen-Code-Ende zu setzen oder die jeweils gewünschte Datei zu laden.

Listing. Select Filename

```
10 TRAP20:GOTO30
20 RESUME NEXT
30 DIRECTORY
40 POKE19,5:INPUT N$
50 POKE19,0:TRAP:PRINT
60 P=INSTR(N$,CHR$(34)):REM 1.ANFUEHRUNGSZEICHEN
65 IF P=0 THEN PRINT"KEIN FILENAME":GOTO110
70 N$=MID$(N$,P+1,LEN(N$)-P): REM ALLES WAS RECHTS DAVON STEHT
80 P=INSTR(N$,CHR$(34)):REM 2.ANFUEHRUNGSZEICHEN
90 N$=LEFT$(N$,P-1):REM ALLES WAS LINKS DAVON STEHT
100 PRINT"FILE "N$" LADEN..."
110 END
```

© 64'er

Listing. Freie Blocks auf Disk

```
0 DIRECTORY"!"
110 POKE19,5
120 POKE1319,13:POKE239,1:REM TASTATURPUFFER MIT RETURN LADEN
130 INPUT"Crs.up";A$:REM CURSOR AUF BLOCKZAHL UND RETURN
135 POKE19,0
140 P=INSTR(A$," ")
150 A$=LEFT$(A$,P) :REM BlÖCKE ALS STRING
160 A=VAL(A$) :REM BLÖCKE ALS ZAHL
```

© 64'er

Freien Platz auf Diskette ermitteln

Auf den ersten Blick ist das gar nicht so einfach, denn es gibt keinen Befehl, mit dem man den freien Speicherplatz einer Diskette enttarnt. Zwar werden nach Durchlauf der Directory-Liste die unbelegten Blocks auf Diskette angegeben, doch wie bekommt man diesen Bereich in eine Variable? Für Basic-Programmierer ist die Auswertung des Blockbelegungsplans (BAM) äußerst kompliziert und dazu noch ausgesprochen langsam.

Mit einem Trick läßt sich aber die angegebene Blockzahl übernehmen. Voraussetzung ist, daß noch zwei freie Zeilen auf dem Bildschirm stehen. Den entsprechenden Programm-Code finden Sie im Listing.

Wenn man vor der Directory-Ausgabe noch die Vorder- und Hintergrundfarbe gleichsetzt, läßt sich die Ermittlung der Blockzahl noch nicht einmal auf dem Screen verfolgen.

INPUT überlistet

Oft ist notwendig, in einer Zeichenkette (String) zu Kommas und Doppelpunkte zu übergeben – der INPUT-Befehl macht da aber nicht mit und ignoriert die genannten Zeichen. Man kann ihn aber leicht überlisten, wenn man den String in Anführungszeichen setzt und weitere Buchstaben oder Zahlen ganz normal eingibt.

Hardware-Tip

Vorgestellt wurde er ausgiebig in der 64'er 6/95: der Expansionsport. Leider fehlt ihm ein I/O-Signal, an das sich eine VIA 6522 oder CIA 6526 anschließen läßt. Allerdings: es sind noch drei Pins frei und die PLA gibt sogar ein I/O-Signal ab. Ursprünglich war das für einen Custom-Chip zur Sprachausgabe geplant – es selektiert den Bereich $FD20 bis-$FD2F.

Man legt einfach eine Brücke aus dünnem Draht von der PLA (U19 im Plus/4, U16 im C16, U101 im C116) Pin 18 an Pin Z des Expansionsports. Ein weiteres Problem (das unübliche Raster von 1,98 mm am Expansionsport) läßt sich lösen, wenn man den seit geraumer Zeit erhältlichen Expansionsport-Verteiler benutzt. Der besitzt nämlich drei 50polige Slots mit gleicher Pinbelegung, aber im 2,54 mm-Raster.

Christian Schäffner

Sortier-Routinen

Nichts geht über Maschinensprache!

In den letzten beiden 64'er-Ausgaben haben wir uns im Sortier-Kurses ausgiebig mit entsprechenden Algorithmen beschäftigt. Diesmal wollen wir dem Geschwindigkeitswunder "Quicksort" mit Maschinensprache-Power noch mehr Beine machen!

Unsere beiden Quicksort-Routinen sind komplett in Assembler speziell für den C 64 und C 128 geschrieben und können durch Basic-Programmierer benutzt werden.

Quicksort 64

In Basic konnte der Quicksort-Algorithmus alle anderen Verfahren abhängen. Mit dem Programm "QUICKSORT.OBJ" auf der Diskette zum Heft legt der C 64 noch einmal zu. Die Routine nutzt den Speicherbereich von 52000 (hex. $CB20) bis 53247 (hex. $CBFF). Dabei belegt das eigentliche Programm nur die Adressen von 52000 bis 52711. Die Speicherstellen dahinter benötigt die Routine als Zwischenspeicher und Programmstack.

"QUICKSORT.OBJ" wird mit SYS 52000 aufgerufen, wobei noch einige wichtige Regeln zu beachten sind:

❶ Das zu sortierende Feld muß ein String-Array sein,

❷ Das Feld muß als erstes im Basic-Programm definiert werden.

❸ Die Speicherstellen 178 (hex. $B2) bis 184 (hex. $B8) und 251 (hex. $FB) bis 254 (hex. $FE) dürfen nicht durch andere Programme benutzt werden. Die Sortierroutine rettet aber die Speicherstellen 178 bis 184 und schreibt die Inhalte nach dem Sortieren zurück.

Das Basic-Listing "QUICKTESTER" auf der Diskette zum Heft zeigt, wie man das Maschinenprogramm installiert und anspricht.

Interessierte Leser finden außerdem den kompletten Quellcode zu "QUICKSORT.OBJ" im Hypra-Ass-Format auf Diskette. Er kann in den Basic-Speicher geladen und wie gewohnt geLISTet werden.

Quicksort 128

Die spezielle C-128-Routine läuft im 40- und 80-Zeichenmodus und ist eine modifizierte Version von "Quicksort-M" (veröffentlicht in der 64'er 7/86). Sie sortiert im FAST-Modus 1000 Strings in knapp anderthalb Sekunden und belegt lediglich 422 Bytes in Bank 1.

Die Ladeanweisung:

```
POKE 57,89: POKE 58,253: CLR:
BLOAD "QUICKSORT 128",ONB1
```

transferiert das Programm ans Ende des Variablenspeichers ($FD59 bis $FEFE) in Bank 1. Die beiden POKE-Anweisungen schützen den belegten Bereich vor Überschreiben, "CLR" paßt bestimmte Zeiger in der Zeropage an. Vorsicht! Damit löscht man automatisch alle Variablenwerte – deshalb sollte man "Quicksort 128" unbedingt am Anfang eines Basic-Programms laden.

Den Sortiervorgang startet man mit der Anweisung:

```
BANK 1: SYS 64857
```

Sortiert wird stets das erste Array eines Programms, das per DIM-Anweisung definiert wurde (z.B. A$(I)). Element 0 (also A$(0)) wird dabei nicht berücksichtigt: man kann dort beliebige Daten ablegen (z.B. Titel der Datei, Programmhinweis usw.).

Unser Demoprogramm "Quicksort-Test" zeigt, wie's funktioniert: zwei Zeichen große Zufallsstrings werden blitzschnell sortiert und ausgegeben.

Thomas Klein,lb,bl

TabCalc C 128 V3.2

Zahlen am laufenden Band

Schneller Überblick zur Aufteilung finanzieller Budgets, Zahlen im Vergleich: das sind wichtige Kriterien, die unser Anwendungsprogramm für den C 128 erfüllt.

Tabellenkalkulationen stellen ein Bildschirmarbeitsblatt zur Verfügung ("Spreadsheet"), auf dem sich beliebige Listen bzw. Zahlenwerte erfassen lassen. Es gliedert sich in Spalten und Zeilen. Jedes Feld kann einen festen Wert, eine Formel oder Text enthalten. Werte und Formeln müssen sich beliebig miteinander verknüpfen lassen.

Unsere Tabellenkalkulation läuft ausschließlich im 80-Zeichenmodus des C 128.

Nach dem Laden und Starten mit der Anweisung:

```
RUN "TAB.KALC.V3.2"
```

holt das Programm die restlichen Dateien von Diskette in den Speicher (Tab.Font, Tab,Syst., Tab.Print und geänderten Zeichensatz), dann werden Sie nach dem aktuellen Datum und dem User-Namen gefragt. Anschließend erscheint ein Untermenü mit folgenden Auswahlpunkten:

❑ **Tabelle laden:** Nach Tipp auf <1> wird der Bildschirm gelöscht, alle Tabellen-Files (Präfix "TAB.") erscheinen auf dem Screen. Geben Sie den gewünschten Dateinamen ein und drücken Sie <RETURN>. Jetzt erscheint das Spreadsheet mit den entsprechenden Daten auf dem Bildschirm.

❑ **Tabelle erstellen:** Damit lassen sich 18 Eingabefelder mit den jeweiligen Datenfeldnamen einrichten (z.B. Gehalt, Miete, Tankkosten, Telefon usw.). Die vorgegebenen Einträge kann man per Tastenkombination <SHIFT CLEAR/HOME> pro Datenfeld löschen oder überschreiben, <RETURN> bringt Sie in die nächste Eingabezeile. Ist die Prozedur beendet, lassen sich falsche Einträge bei Bedarf ändern.

Jetzt ist die Definition der horizontalen Spaltenbezeichnungen an der Reihe (Vorgabe: Monatsnamen). Man kann jedoch jeden beliebigen Bezeichnungstext eintragen. Anschließend wird das noch leere Spreadsheet nach einer Abfrage auf Disk gespeichert.

Zahleneingabe: Hauptaugenmerk jeder Tabellenkalkulation ist das Jonglieren mit berechenbaren Zahlen und Werten – nach Tipp auf <RETURN> verschwindet der untere Info-Balken: jetzt lassen sich gewünschte Zahlen eingeben (Maximalwerte zwischen -9999.99 bis 9999.99 pro Spalte/Zeile).

Leider sind aus Platzmangel auf dem Screen keine größeren Zahlen möglich; auch Berechungsergebnisse müssen sich an dieses Limit halten.

Werte mit Nullen hinter dem Komma darf man als Ganzzahl angeben (z.B. "23" statt "23.00") - das Programm setzt die Nachkommastellen automatisch. Per <RETURN> sind die Eingaben zu bestätigen. Die aktuellen Summen der Zeilen (rechts außen) und Spalten (unten) erscheinen automatisch, auch die Effektivsumme wird je nach Eintrag angepaßt (dort sind natürlich Werte möglich, die den Betrag von 9999.99 übersteigen). <RETURN> ohne Zahleneingabe beendet die Arbeit im Spreadsheet

Die Funktionstasten des C 128 erfüllen nützliche Aufgaben bei der Programmarbeit (s. Tabelle).

Integrierte Taschenrechnerfunktion

Nach Tipp auf die ESC-Taste öffnet sich der Rechner-Screen. Möglich sind die vier Grundrechenarten und Prozentrechnung. Das Ergebnis wird nach jedem Rechenvorgang im RAM abgelegt und erscheint nach Tipp auf <F3>. Anschließend kann man den Wert per <F3> unverändert ans Spreadsheet übergeben.

Der integrierte Rechner funktioniert wie jeder normale Taschenrechner: die erste Eingabe ist nicht mit <RETURN> zu bestätigen, sondern mit der Taste, die die nachfolgende Rechenfunktion repräsentiert (also <*>, <+>, <->, <:> usw). Beachten Sie dabei, daß die Tabellenkalkulation mit der DIN-Tastenbelegung arbeitet und bei Divisionen nicht der Schrägstrich, sondern ein Doppelpunkt zu verwenden ist.

Prozentrechnung: Anders als bei üblichen Taschenrechnern sind die Eingaben im Denk- bzw. Schreibrhythmus auszuführen.

Beispiel Prozentrechnung 1:

Wie hoch ist der Bruttobetrag aus 350,86 Mark plus 15 Prozent Mehrwertsteuer?

Eingabe: 350.86
Taste: <+>
Eingabe: 15
Taste: <%>
Taste: <=> oder <RETURN>
Summe: 403.49

Das Ergebnis (Summe) wird ebenfalls unter <F3> gespeichert. Bei den übrigen Grundrechenarten ist die Funktionsweise identisch: der eingetragene Prozentwert wird vom Grundwert ermittelt und hinzuaddiert, subtrahiert, dividiert oder multipliziert.

Beispiel Prozentrechnung 2:

Wie hoch ist der Mehrwertsteuerbetrag (15 Prozent), der in 403,49 Mark enthalten ist?

Eingabe: 403.49
Taste: <%>
Eingabe: 15
Taste: <=> oder <RETURN>
Summe: 52.63

Rechnen Sie nach: 403,49 minus 52,63 ergibt wieder 350,86. Das Ergebnis wird ebenfalls im <F3> gesichert. Es läßt sich wie gewohnt ins Spreadsheet Dazu ein Hinweis: <F3> kann stets nur einen einzigen Wert enthalten – jeder neue überschreibt den alten. Achtung: Höhere Zahlen als 9999.99 werden ignoriert und nicht übernommen – dann ist bei <F3> kein Wert gespeichert.

Diethelm Kretschmann/bl

31.12.1995 — Interact Software

	JAN	FEB	MAR
Einkommen...	1500.00	1650.00	1550.00
Steuerrueckz	100.00	111.11	0.00
Dividende...	100.00	100.00	0.00
Gewinnsparen	0.00	0.00	0.00
Lebensunterh	-920.00	-900.00	-930.00
Miete.......	-250.00	-250.00	-250.00
Tanken......	-100.00	-100.00	-120.00
Telefon.....	-35.00	-40.00	-30.00
Abtragung...	0.00	0.00	0.00
Taschengeld.	0.00	0.00	0.00
Rundfunkgeb.	0.00	0.00	0.00
Autosteuer..	-250.00	0.00	0.00
Autovers....	-350.00	0.00	0.00
Rechtschutz.	-150.00	0.00	0.00
Sparen......	0.00	0.00	0.00
ADAC........	0.00	0.00	0.00
Nebenkosten.	0.00	-200.00	0.00
sonstiges...	-6.00	2.00	3.00
Summe:	-361.00	373.11	223.00

Eingabe=RETURN F1=Blattern F5=Druck
ESC=Rechner F3=Rechn

Beispiel-Spreadsheet zur Tabellenkalkulation: maximal 18 Datenfelder passen auf den VDC-Screen

TabCalc 128 - Funktionstastenbelegung

Taste	Aufgabe
<F1>	Wechsel zwischen Bildschirm 1 und 2
<F3>	Rechnersumme (gibt das Rechenergebnis des im Programm integrierten "Taschenrechners" aus). Die Zahl läßt sich unverändert ins Spreadsheet übernehmen (Eingabe-Funktion per <RETURN> aktivieren und F3 drücken).
<F5>	Spreadsheet läßt sich auf Epson-kompatiblen Druckern ausgeben. Falls das Gerät nicht "online" ist, erscheint eine Fehlermeldung.
<F6>	... sichert die Tabelle mit entsprechendem Dateinamen auf die aktuelle Diskette.im Laufwerk.
<F7>	... lädt ein anderes Spreadsheet von Disk. Vorher erscheint eine Sicherheitsabfrage, ob die aktuelle Tabelle auf dem Screen gespeichert werden soll. Falls man die Frage bejaht, wird erneut das Hauptmenü aktiviert, dessen Funktion uneingeschränkt zur Verfügung stehen.
<F8>	Programmaustieg per Reset. Wurde im aktuellen Spreadsheet auf dem Screen etwas geändert, ruft das Programm vorher die Speicherfunktion auf.
<ESC>	... aktiviert die Taschenrechner-Funktion

Mathematik

Der Geometrie-Profi

Mit dem "Geometrie Profi" berechnen Sie im Handumdrehen Körper und Flächen. Die grafische Benutzeroberfläche und über 230 integrierten Formeln machen das Programm zu einem fabelhaften Handwerkszeug für Hobby-Mathematiker und Schüler.

Das Programm "Geometrie-Profi" wird mit einem Joystick in Port 2 oder einer Maus in Port 1 gesteuert. Nach dem Laden mit:

```
LOAD"GEOMETRIE-PROFI",8,1
```

und dem Start mit dem RUN-Befehl, erscheint das Titelbild. Es wird mit der SPACE-Taste verlassen und das Hauptmenü eingeblendet. Hier wählt man zwischen Körper- oder Flächenberechnung. Dazu klicken Sie mit dem Mauszeiger das betreffende Feld auf dem Bildschirm an.

In der Flächen- bzw. Körper-Auswahl läßt sich nun das gewünschte Objekt wählen. Jetzt kommt man in den Berechnungsschirm. Hier wird das aktuelle Objekt dargestellt und das Programm wartet auf die Eingabe der Werte.

Im Rechen-Menü finden Sie die Programm-Optionen:

FLÄCHE/KÖRPER: Wechsel zwischen Flächen- und Körper-Menü

WECHSEL: Neues Objekt für die Berechnung wählen

NEU: Alle Daten auf dem Bildschirm und im Speicher löschen

START: Beginn der Berechnung

Um Werte einzugeben, klicken Sie einfach mit dem Mauszeiger auf das Variablenfeld und übergeben mit der Tastatur die Zahlen. Fehlen Werte zur Berechnung, verweigert das Programm die Mitarbeit.

Um vernünftige Ergebnisse bei der Berechnung zu erhalten gelten die Bedingung in Tabelle 1. Tabelle 2 zeigt die Definition der Flächen und Tabelle 3 die Abkürzungen, die das Programm in den Eingabefeldern verwendet.

Daniel Schulte/lb

Tabelle 1: Berechnungsbedingungen "Geometrie-Profi"

Name	Bedinung
Flächen	
Rechteck	e=>a, b u=>a, b, e
Rhombus	u=>e, f
Trapez	u=>a, b, c, d, m, h h=b, d
Parallelogramm	e=>a, b f=>a, b h=b, e, f
Sehnen-Viereck	u=>a, b, c, d, e, f
Drachenviereck	u=>a, c, e, f
Vieleck	u=>a
Kreis	u=>r
allgemeines Dreieck	u=> a, b, c, hc
rechtwinkliges Dreieck	u=>a, b, c, p, q, hc
gleichseitiges Dreieck	u=>a, h
Körper	**Bedeutung**
Hohlzylinder	r2=>r1, a
Kreiskegel	s=>h
Kegelstumpf	r1=>r2 s=>h
Pyramidenstumpf	Ag=>Ad Ao=Am Am=>A1
Kugelabschnitt	r=>h,R
Kugelausschnitt	r=>h,R
Kugelschicht	r=>R1, R2 2r=>h

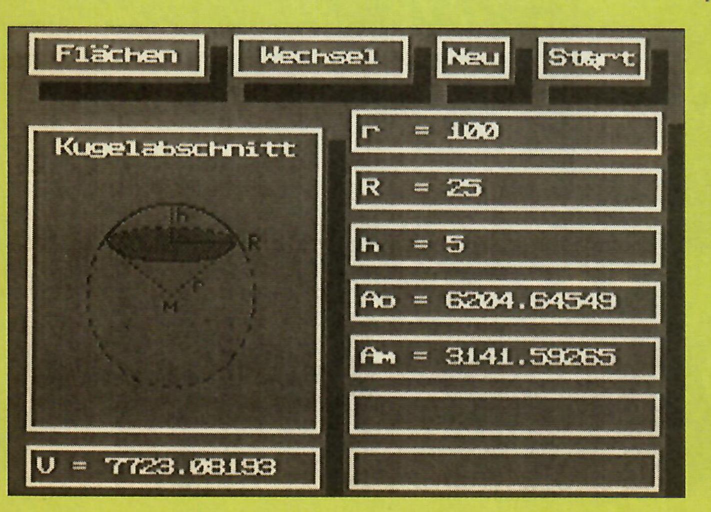
Tabelle 3: Die verwendeten Abkürzungen

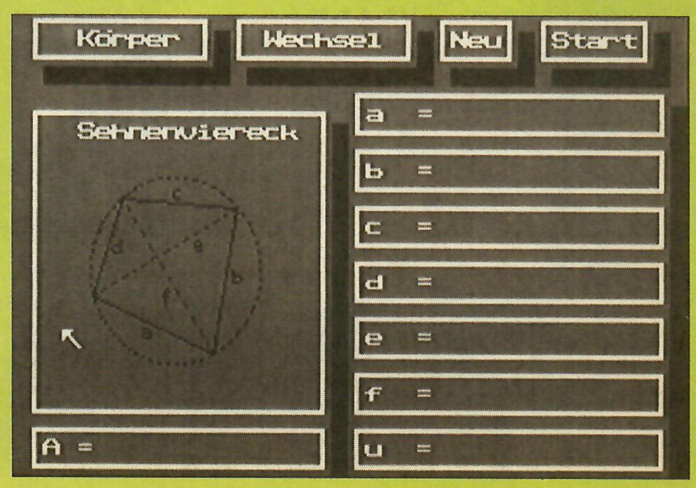
Bezeichnung	Bedeutung
V	Volumen
Ad	Deckfläche
Ag	Grundfläche
Am	Mantelfläche
Ao	Oberfläche
a,b,c,d	Kantenlängen
s	Seitenkante
h	Körperhöhe
e	Diagonale

Tabelle 2: Flächen-Definitionen

Fläche	Definition
Rechteck	- alle Innenwinkel sind 90 Grad - Diagonalen sind gleich lang - gegenüberliegende Seiten sind parallel und gleich lang
Rhombus (Raute)	- die Diagonalen stehen senkrecht aufeinander und halbieren sich - alle Seiten sind gleich lang - gegenüberliegende Seiten sind parallel
Trapez	- mindestens zwei Seiten sind parallel
Parallelogramm	- die Diagonalen halbieren einander - die gegenüberliegenden Winkel sind gleich groß - gegenüberliegende Seiten sind parallel und gleich lang
Drachenviereck	- die Diagonalen stehen senkrecht aufeinander - mindesten zwei gegenüberliegende Winkel sind gleich groß
Sehnenviereck	- alle Eckpunkte liegen auf einem Kreis - die Summe der gegenüberliegenden Winkel ist 180 Grad
regelmäßiges Vieleck	- alle Innenwinkel sind gleich groß - alle Seiten sind gleich lang

Der Geometrie-Profi wird wahlweise mit Maus oder Joystick bedient





Quadra

Man werfe "Tetris" und "Vier gewinnt" in einen Topf, gebe einige Überraschungen in Form von Extras bei und würze die Mixtur mit tollen Grafiken und Sounds - fertig ist "Quadrant". Genau das richtige Menü für den Spiele-Feinschmecker und Tüftel-Freak!

Bevor Sie in den Genuß unseres Denkspiel-Potpouris kommen, legen Sie die Seite 2 der Heftdiskette ein und laden Sie das Game mit:

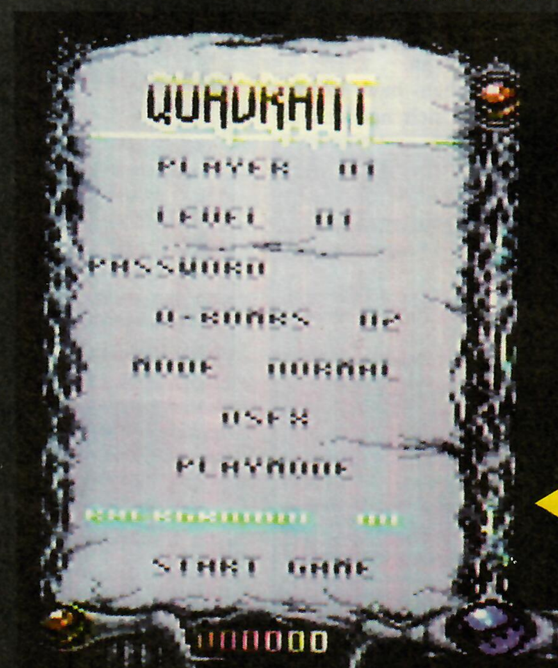
```
LOAD"QUADRANT LOADER",8,1
```

Es wird nun mit dem RUN-Befehl gestartet. Während ein Intro Spiel und Programmier-Team vorstellt, wird das Hauptprogramm nachgeladen. Hat die Floppy ihre Arbeit verrichtet, geht es per Feuerbutton ins Hauptmenü des Spiels. Auf der linken Seite dieses Bildschirms können Sie die Spieloptionen einstellen:

PLAYER: Wahl zwischen Ein- und Zwei-Spieler-Mode

LEVEL: Start-Abschnitt festlegen

PASSWORD: Zugangscode für eingestelltes Level eingeben (Auswahl mit dem Joystick, Bestätigung mit Feuerbutton)

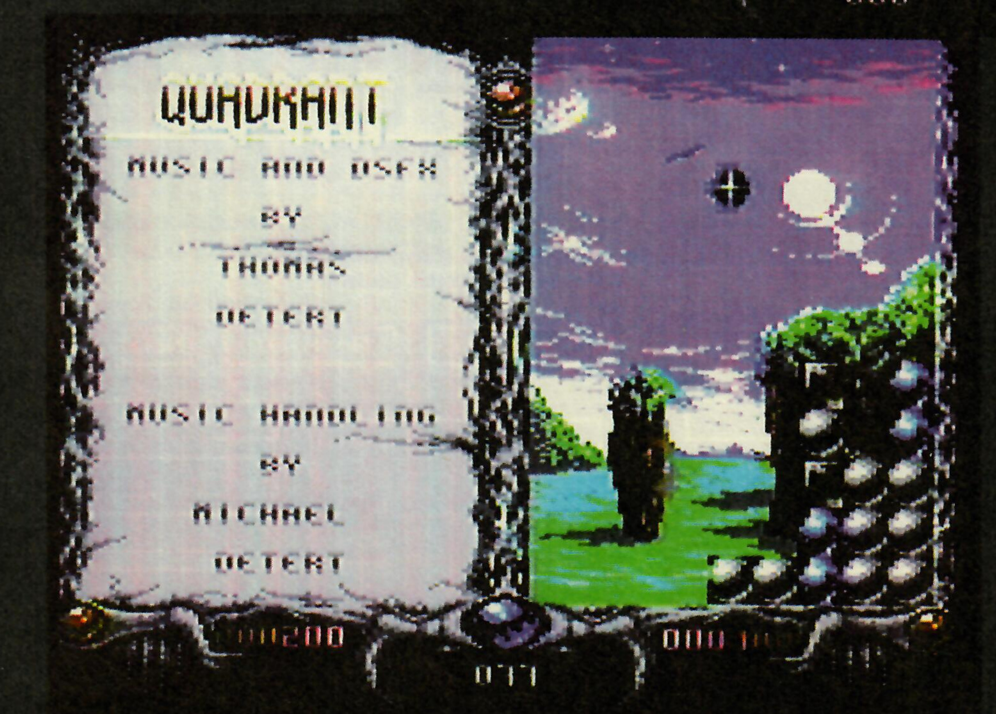


Wichtiger Hinweis

Wenn Sie sich eine Sicherheitskopie des Spiels machen wollen, müssen Sie unbedingt ein Backup-Programm benutzen, das ganze Diskettenseiten kopiert. In keinem Falle ein Filecopy verwenden oder die Diskette per Validate-Befehl bearbeiten. Wenn Sie das Game von der Diskette zu Heft spielen wollen, müssen Sie für Seite 2 unbedingt die Schreibschutzkerbe einstanzen.

Nach der Wahl der Spieloptionen und der Eingabe des(r) Spielernamen geht es sofort ins erste Level. Mit einem Paßwort können Sie in höheren Ebenen einsteigen.





Q-BOMBS: Anzahl der Bomben zum Sprengen von Hindernissen

MODE: Schwierigkeitsgad des Spiels einstellen

DSFX/MUSIC: Musik oder Sound-Effekte wählen

PLAY-/DEMO-MODE: Wechsel zwischen Spiel und Demonstration von Quadrant

BACKGROUND: Hintergrund-Grafik an- bzw. abschalten

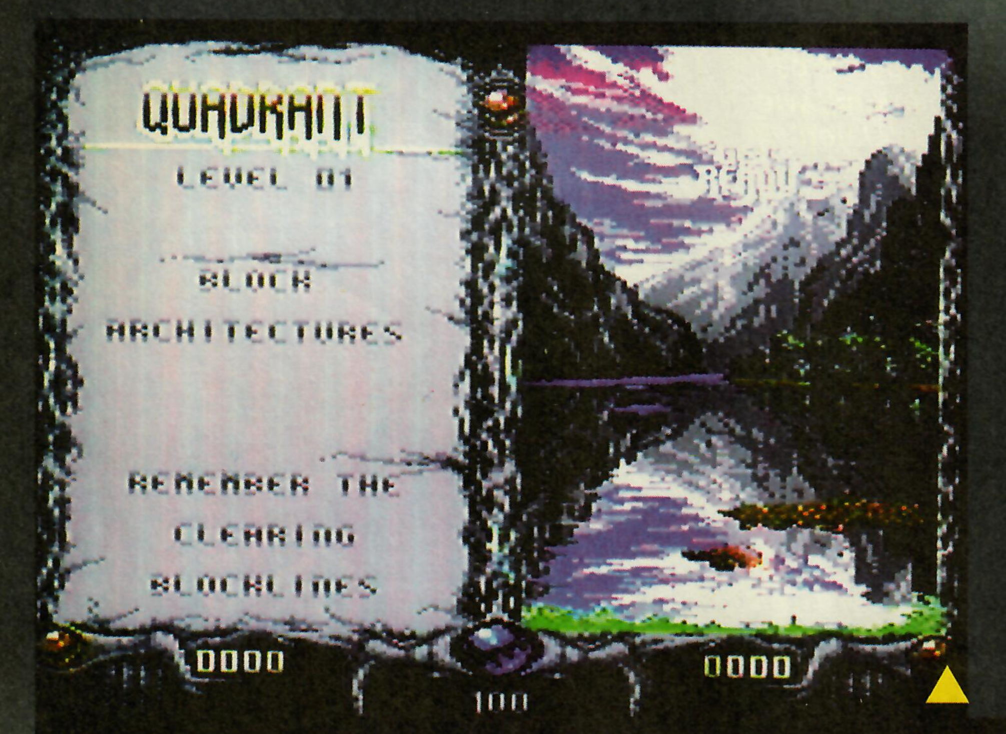
START GAME: Spielbeginn

Die Eingabe der Spieler-Namen erfolgt per Joystick, wobei rechts/links den Cursor positioniert, auf/ab in der Buchstabenliste blättert. Der Feuerknopf beendet die Eingabe.

Im Spiel fallen abwechselnd blaue Kristalle und graue Kugeln in einen Behälter. Sie müssen so

nt

Quadrantt



plaziert werden, daß immer vier gleiche Objekte eine horizontale, vertikale oder diagonal Linie ergeben. Ist so eine Linie komplett, lösen sich die betreffenden Spielelemente auf und der restliche Stapel rutscht nach. Der Spieler bekommt dafür Punkte auf dem High-Score-Konto gutgeschrieben. Natürlich können sich die Spielpartner gegenseitig die Suppe versalzen und fast fertige Kombinationen durch eigene Steine blockieren. Der Behälter füllt sich dadurch immer mehr und droht überzulaufen. In diesem Falle erscheint unausweichlich der Game-Over-Screen und das Spiel ist aus...

Als Überraschung fallen (zufallsgesteuert) Steine in den Behälter, die sich entweder als Blocker entpuppen oder ein Extra enthalten. Die Blocker vermiesen zu den unmöglichsten Zeitpunkten das Positieren wichtiger Steine. Die Extras sorgen für unterschiedliche Effekte:

- manchmal verwandeln sich die Spielobjekte in ihre Gegenstücke und die hart erarbeiteten Kombinationen fallen dem Gegenspieler zu
- wenn es der Zufall will, werden die Steine im Spielfeld unsichtbar
- oder die Joysticksteuerung reagiert in umgekehrter Richtung...

Sind genügend Vierer-Kombinationen aufgebaut, räumt das Spiel den Behälter leer und lädt die nächste Spielstufe automatisch nach. *Jörn-Erik Burkert*

Die Quadrant-Extras entpuppen sich mal als Hilfen oder vermiesen auch einmal das Tüftelvergnügen! Dann erscheint schnell der Bildschirm mit "Game Over"...



Auf Diskette im Heft

Workshop: Dateiverwaltung

Die meist verbreitete Anwendung für den C 64 ist die Verwaltung von Datensammlungen jeglicher Art: Videos, CDs, Briefmarken, Bücher usw. Wir zeigen Ihnen in unserem mehrteiligen Programmierkurs, wie man man Daten effektiv erfaßt, pflegt und die Techniken des C-64-Betriebssystems sinnvoll einsetzt.

Folge 3

Datenbank GmbH

Daten sind auf Diskette zwar sicher gespeichert, aber gar nichts wert, wenn man sie nicht wieder sichtbar machen kann: auf dem Screen oder per Druckausgabe.

Das erledigen weitere Unterprogramme, die man unserem Software-Fragment (s. Listings in der 64'er 6/95) hinzufügen muß:

Daten lesen: Das Flag in Zeile 8000 gibt Auskunft darüber, ob sich eine REL-Datei auf Disk befindet oder noch gar nicht geöffnet wurde. Dann gibt man den Suchbegriff ein, der in der letzten Kursfolge als ID$(AD) in der indexsequentiellen Datei definiert wurde (s. Listing "Daten eingeben"). Drei Datensätze finden Sie in der REL-Datei "Adressen". Die entsprechenden Suchbegriffe: *GUC, CMD, PPE.*

Wenn der Computer fündig wird, positioniert das Programm den entdeckten Datensatz und holt ihn in den Speicher (Zeilen 8100 bis 8110). Andernfalls erscheint eine Fehlermeldung und das Programm kehrt zum Hauptmenü zurück.

Gesamtstring aufteilen: Die Eingaben zu Datensätzen von REL-Files speichert der Computer hintereinander ab – quasi in einer langen Zeichenkette: die zusammengefaßten Inhalte der jeweiligen Datenfelder. Um sie ausgabefähig zu machen, sind die jeweiligen Datenfelder daraus wieder in passender Länge zu generieren – darum kümmern sich die Zeilen 8130 bis 8138 in unserer Ausgaberoutine.

Ausgabe auf Screen: Nachdem die einzelnen Datenfelder jetzt wieder als Variablenwerte existieren (DS$(C)), ist es nur noch ein Kinderspiel, sie auf den Bildschirm zu bringen: das übernimmt das entsprechende Unterprogramm (s. Listing).

Um der ganzen Angelegenheit einen professionelleren Touch zu geben, greift man auf die Subroutine "Window" zurück (s. Listing, Zeilen 9900 bis 9920), die auf dem Screen ein Ausgabefenster für den Datensatz erzeugt. Je nach Umfang der Datenfelder läßt es sich vergrößern oder verkleinern (Wert der Variablen C in der FOR_NEXT-Schleife , Zeile 9907) erhöhen bzw reduzieren).

Was man schwarz auf weiß besitzt ...

Als Datei auf Disk oder im Computer-RAM nützen die Datensätze wenig: falls man eine Adresse sucht, muß man stets den C 64 aktivieren, das File laden usw. Dateiverwaltungen sind also ohne Druckroutine nur die Hälfte wert. Darum kümmert sich unser Unterprogramm "Druckausgabe" (s. Listing). In der neuen Version unseres Demoprogramms "Reldat V2.0" ist es bereits enthalten.

Die Druckroutine ist auf seriell angeschlossene Printer (evtl. mit Hardware-Interface) zugeschnitten und ähnelt im Programmablauf dem Unterprogramm für die Bildschirmausgabe.

Allerdings fehlt uns im gesamten Hauptprogramm der entsprechende Schalter (also eine Abfrage), um die Druckausgabe zu realisieren. Das läßt sich am besten unmittelbar nach der Screenanzeige realisieren, deshalb ist die Abfrage dort einzubauen.

Ändern Sie deshalb das Listing von "Reldat V1.0" (Programmservice-Disk 64'er 6/95):

```
8180 gosub 8300
```

Zeile 8182 ist ersatzlos zu streichen. Das Unterprogramm ab Zeile 8300 (Druckroutine) finden Sie im entsprechenden Listing, in der neuen Version von Reldat sind diese Programmzeilen ebenfalls enthalten. Per Taste läßt sich die Druckausgabe aktivieren, anschließend kehrt das Programm zum Hauptmenü zurück.

Wichtiger Hinweis: die Nummer für den Datenkanal zum Drucker muß sich unbedingt von der Zahl unterscheiden, die zum Öffnen des Kanals unserer relativen Dateiverwaltung verwendet wird (hier: "1"). Die Druckausgabe verwendet die im C-64-Betriebssystem integrierten seriellen Datenübertragungsfunktionen – deshalb spielen nur Printer mit, die seriell mit dem C 64 verbunden sind (per DIN-Kabel oder Hardware-Interface).

Datensätze ändern

Adressen oder beliebige Datensammlungen haben die unangenehme Eigenschaft, sich nach absehbarer Zeit zu ändern. Ein Kunde zieht um, eine Videokassette wird neu bespielt, der beste Freund bekommt eine siebenstellige Telefonnummer – müßte man jetzt wegen kleiner Änderungen den Datensatz jedesmal neu erfassen, würde man bald die Lust verlieren.

Deshalb ist in unser Programmprojekt noch eine Editier-Routine einzubauen, die den jeweiligen Inhalt des Records auf den Screen bringt und Änderungen am Eingabetext direkt zuläßt, ohne den alten Inhalt vollständig zu löschen. Darum werden wir uns in der nächsten Folge unseres Dateiverwaltungskurses kümmern.

Harald Beiler

Listing. Daten ausgeben

```
997 REM *************************
7998 REM *   DATEN AUSGEBEN      *
7999 REM *************************
8000 IF FL=0 THEN GOSUB 9500 :RETURN
8010 PRINTCHR$(147):POKE19,64:INPUT"INDEX-
BEGR.:";IX$:POKE19,0:REM INDEXABFRAGE
8020 FOR C=1 TO AD: REM INDEXDATEI DURCHSUCHEN
8030 IF IX$=ID$(C) THEN GN=C:C=AD:NEXT C:GOTO 8100: REM
INDEX GEFUNDEN
8040 NEXT C
8050 PRINTCHR$(147)"SCHLUESSELWORT FALSCH
ODER":PRINT"NICHT VORHANDEN!"
8052 FORT=1 TO 1000:NEXT
8060 PRINTCHR$(147):GOSUB9900:RETURN
8100 RN=GN:RP=1:RC$="":GOSUB2000:REM RECORD POSTIONIEREN
8110 INPUT#LF,RC$: REM RECORD HOLEN
8119 REM ***************************
8120 REM * GESAMTSTRING AUFTEILEN *
8121 REM ***************************
8130 DS$(1)=MID$(RC$,1,15): REM NACHNAME
8132 DS$(2)=MID$(RC$,16,10): REM VORNAME
8134 DS$(3)=MID$(RC$,26,15): REM STRASSE
8136 DS$(4)=MID$(RC$,41,5): REM PLZ
8137 DS$(5)=MID$(RC$,46,15): REM WOHNORT
8138 DS$(6)=MID$(RC$,61,11): REM TELEFON
8139 REM *****************************
8140 REM DATENSATZ AUF SCREEN AUSGEBEN
8141 REM *****************************
8142 PRINTCHR$(147)LEFT$(LA$,79)
8143 PRINTCHR$(19)CHR$(18)TAB(2)" dATENAUSGABE
"TAB(23)"RECORD-NR.: ";RN
8150 PRINTCHR$(13):FOR C=1 TO 6
8160 PRINTTAB(5)DS$(C)
8170 NEXT C
8172 PRINT:PRINTLEFT$(LA$,79)+CHR$(145)+CHR$(145)
8180 PRINT:PRINTTAB(16)CHR$(18)" TASTE! "
8182 POKE198,0:WAIT198,1
8185 PRINTCHR$(147):GOSUB
9900:RETURN
9497 REM **************************
9498 REM * KEINE DATEI GEOEFFNET! *
9499 REM **************************
9500 PRINTCHR$(147):GOSUB 9900
9530 PRINTCHR$(19)CHR$(13)CHR$(13)TAB(13)" FEHLERMELDUNG
"CHR$(13)CHR$(13)
9535 PRINTTAB(12)"DATEI WURDE NOCH"
9536 PRINTTAB(12)"NICHT INITIALISIERT"
9537 PRINTTAB(12)"ODER ANGELEGT!!!"
9590 FORT=1TO1500:NEXT
9592 PRINTCHR$(147):GOSUB9900
9599 RETURN
9897 REM *************************
9898 REM * UNTERPR. WINDOW       *
9899 REM *************************
9900 E1$=CHR$(18)+CHR$(176):E2$=CHR$(18)+CHR$(174)
9901 E3$=CHR$(18)+CHR$(173):E4$=CHR$(18)+CHR$(189)
9902 UL$=CHR$(18)+CHR$(171):UR$=CHR$(18)+CHR$(179)
9903 LN$=""
9904 FORI=1TO22:LN$=LN$+CHR$(18)+CHR$(192):NEXT
9905 LA$=LN$+LN$
9906 PRINTCHR$(19):PRINTTAB(9)E1$LN$E2$
9907 ZL=2:SP=9:GOSUB7900
9908 FOR C=1 TO 7
9910
PRINTTAB(9)CHR$(18)CHR$(221)TAB(32)CHR$(18)CHR$(221)
9911 NEXT C
9913 PRINTTAB(9)E3$LN$E4$
9915 ZL=3:SP=9:GOSUB7900
9916 PRINTTAB(9)UL$LN$UR$
9920 RETURN
```

Listing. Druckausgabe der Records

```
8195 REM **********************
8196 REM *  DRUCKAUSGABE      *
8197 REM **********************
8200 OPEN1,4,7:CMD1
8210 PRINT#1,"RECORD-NR.:";STR$(RN)
8220 FOR C=1 TO 6
8230 PRINT#1,DS$(C)
8240 NEXT C
8250 PRINT#1: CLOSE 1
8260 RETURN
8300 PRINT:PRINTTAB(8)CHR$(18)" DATENSATZ DRUCKEN (J/N)
"
8310 GETT$
8320 IF T$="J" THEN GOSUB 8200: RETURN
8330 IF T$="N" THEN RETURN
8340 GOTO 8310
```


Dieses Symbol zeigt an, welche Programme auf Diskette erhältlich sind

Geos-Systemroutinen

Geos intern

Folge 6

Die Systemdatei "Geos Kernel", Dreh- und Angelpunkt der beliebten C-64/C-128-Benutzeroberfläche, haben wir schon in ihre Bestandteile zerlegt – jede Menge phantastischer Assembler-Routinen kamen dabei zum Vorschein. Heute zeigen wir Ihnen, wie Geos den I/O-Bereich des C 64 behandelt und beamen uns in die Welt der komfortablen Geos-Diskettenroutinen.

Geos-Kernel enthält acht raffinierte Unterprogramme, die den sensiblen I/O-Bereich des Computers (ab $D000) effektvoll manipulieren. Außerdem greift man mit diesen Routinen auch aufs Floppy-RAM sowie den seriellen Anschluß und den Userport zu.

Weitere 59 komfortable Unterprogramme zur Manipulation von Floppylaufwerken und Disketten repräsentieren den Hauptteil von Geos-Kernel – sie nehmen dem Anwender jede Menge Arbeit ab.

Man unterteilt sie in vier funktionelle Klassen:

❶ **High-Level-Routinen:** ... beziehen sich auf die gesamte Diskette oder vollständige Files beliebigen Typs,

❷ **Mid-Level-Unterprogramme:** ... weisen bestimmten Dateien entsprechende Aufgaben zu,

❸ **Low-Level-Routinen:** ... manipulieren einzelne Blöcke auf Disk, allerdings keine zusammenhängenden Files.

❹ **VLIR:** ... ein Routinentyp, der sich ausschließlich mit der eigens vom Geos-Hersteller entwickelten speziellen Art der Dateiverwaltung befaßt. *bl*

```
6.7.88          GeoMonitor V2.1          13:11
GEOS-Version: 2.0        Seriennr.: 12397

Eingabe: JOYSTICK     Ausgabe: TurboDriver
Laufwerk 1:  1541   Disk: 64'ER-SONDERH.88   Turbo: aktiv
Laufwerk 2:  ????   Disk:      Turbo: keine
Laufwerk 3:  ????   Disk:      Turbo: keine
Laufwerk 4:  ????   Disk:      Turbo: keine
RAM-Erweiterung: 0 KByte     akt. Geräteadresse: 8

                    Vektoren:
IRQ          BRK      NMI           KRNL    RESET
$94fe/$fab3  $fe66    $9503/$c000   $f34a   $fb35

Datum: 6.7.88    Zeit: 13:11:43

GeoMonitor V2.1 vom 4.3.92 18:29

                     OK
```

Alle Geos-Applikationen mit integriertem Diskettenmonitor greifen auf Kernel-Routinen zurück: Beispiel "GeoMonitor V2.1".

InitForIO ($C25C)

Diese Routine bereitet alle Ein- und Ausgaben vor: der Interrupt wird abgeschaltet und das Original-Betriebssystem des C 64 sowie der I/O-Bereich ab $D000 aktiviert. Beim C 128 reduziert sich die Taktfrequenz automatisch auf 1 MHz.
Nach dem Aufruf der Routine lassen sich z.B. die Sprite-Farben setzen. InitForIO wird von den meisten Diskettenroutinen intern verwendet; bei anderen wie ReadBlock, WriteBlock und VerWriteBlock ist InitForIO vorher ins Programm einzubauen. Serielle und parallele Geos-Druckertreiber benutzen InitForIO ebenfalls, um Daten zum jeweiligen Port zu schicken.
Zum Aufruf der Routine sind keine Parameter nötig; verändert werden Akku und y-Register. Beachten Sie, daß der Computer-Interrupt abgeschaltet wird!

DoneWithIO ($C25F)

... macht InitForIO rückgängig und schließt wieder alle Ein- und Ausgabe-Möglichkeiten des I/O-Bereichs: der Interrupt und das Geos-System werden erneut eingeschaltet, beim C 128 wieder die exakte Taktrate (2 MHz) gesetzt.
Die Routine braucht keine Parameter; Akku und y-Register werden verändert.

ChangeDiskDevice ($C2BC)

Diese Routine erwähnen wir lediglich der Vollständigkeit halber, da sie von Programmierern niemals eingesetzt wird. Den Austausch von Laufwerken (z.B. A und C) sollte man stets im Desktop erledigen (damit wechselt man auch gleichzeitig die entsprechende Treiber-Software).

ExitTurbo ($C232)

... schaltet den Turbo-Modus ab. Das macht auch SetDevice nach Beendigung der Routine.
Es werden keine zusätzlichen Parameter benötigt, intern schaltet aber curDrive die Turbosoftware des aktiven Laufwerks ab. Die Fehlernummer steht im x-Register, Akku und y-Register ändern sich.

EnterTurbo ($C214)

... aktiviert die Turbo-Routinen innerhalb der Geos-Laufwerke und lädt sie notfalls ins Floppy-RAM.
Achtung: bei den Diskettenroutinen ReadBlock, WriteBlock, VerWriteBlock und ReadLink muß der Programmierer gewünschte Turbo-Routinen selbst aufrufen!
Parameter entfallen, allerdings greift EnterTurbo intern auf curDrive und curType zurück. Die Fehlernummer steht im x-Register, Akku und y-Register werden verändert.

OpenDisk ($C2A1)

... öffnet die Disk im aktuellen Laufwerk. Intern ruft Geos zusätzlich folgende Routinen auf: NewDisk, GetDirHead, ChkDkGEOS und GetPrtCurDkNm. Aktivieren Sie die Routine jedesmal, wenn eine neue Diskette eingelegt oder ein anderes Laufwerk gewählt wird (nach SetDevice) - sonst gibt's Komplikationen (z.B. Übertragen des BAM-Inhalts im Speicher auf eine andere Disk).Parameter entfallen. Die Fehlerkennzahl steht im x-Register (0 = kein Fehler), in R5 der Zeiger auf den Diskettennamen.

PurgeTurbo ($C235)

... geht im Vergleich zu ExitTurbo noch einen Schritt weiter: die Turbo-Routine wird komplett aus dem Floppy-RAM entfernt (das Original-Commodore-DOS hat wieder Vorrang).
PurgeTurbo wird auch von Desktop aufgerufen, um beispielsweise eine Disk zu formatieren.
Die Routine stützt sich ebenfalls auf curDrive und braucht keine weiteren Parameter.
Im x-Register findet man nach dem Routinen-Durchlauf die Fehlernummer. Geändert werden Akku, y-Register sowie die Inhalte der 16-Bit-Systemregister R0 bis R3.

GetPtrCurDkNm ($C298)

... weist auf einen 16-Bit-Wert (Zeiger), den Sie beliebig in jeder freien Zeropage-Adresse ablegen können: er zeigt auf die Speicherposition des aktuellen Disketten-Namens, der aus maximal 16 Zeichen besteht (kürzere Bezeichnungen werden mit $A0-Werten aufgefüllt).
Die gewählte Zeropage-Adresse für den Zeiger ist als Parameterangabe im x-Register einzutragen.

FirstInit ($C271)

Teil der Initialisierungs-Routine beim Start von Geos. Ein Großteil der Systemvariablen wird auf die Defaultwerte zurückgesetzt. Der Systemzeichensatz ist zusätzlich durch die Routine UseSystemFont zu aktivieren.Parameterangaben entfallen, verändert werden x- und y-Register sowie die Inhalte von R0 bis R2.

SetGEOSDisk ($C1EA)

... verwandelt eine bereits im Normal-Floppy-DOS formatierte Disk ins Geos-Format: im BAM-Block (Spur 18, Sektor 0) wird der Hinweistext "GEOS format V1.0" eingetragen und ein Border-Block erzeugt. Voraussetzung: vorher ist die Routine OpenDisk aufzurufen, anschließend werden intern weitere Kernel-Routinen aktiviert: GetDirHead, CalcBlksFree, SetNextFree (speziell für den Borderblock) und PutDirHead. Parameter sind überflüssig.
Das x-Register enthält die Fehlernummer ($00 = kein Fehler), Akku, y-Register, das Low-Byte von R0 sowie R1, R4 und R5 werden verändert - ebenso die Systemvariablen curDirHead und dir2Head (1571/1581).

FindFTypes ($C23B)

... erzeugt eine Tabelle von File-Namen gleichen Typs (z.B. Applikationen, DeskAccessories, GeoWrite-Texte usw.). Der Dateiname besteht aus 16 Zeichen plus einem Nullbyte.

Parameterangaben:
R7-Low: File-Typ,
R7-High: maximale Anzahl der zu übernehmenden Dateinamen,
R10: Zeiger auf Class ($00 als Endekennzeichen),
R6: Speicheradresse, ab der die gefundenen File-Namen im RAM abgelegt werden.
FindFTypes durchforstet das Directory der aktuellen Disk, vergleicht die File-Typen mit dem Wert in R7L und die Klassifizierung (Class) mit dem Eintrag in R10. Die letztgenannte Funktion ist optional: schreibt man $0000 in R10, verzichtet Geos auf den Vergleich.
Im x-Register wird die Fehlernummer, im High-Byte von R7 die Anzahl der noch verbleibenden Einträge vermerkt.

FindFile ($C20B)

... sucht im Disketteninhaltsverzeichnis nach einer bestimmten Datei, deren File-Namen-Adresse im Systemregister R6 stehen muß. Der File-Name sollte als Endekennzeichen unbedingt ein Nullbyte enthalten! Wird die Routine nicht fündig, trägt Geos die Fehlernummer $05 (File not found) ins x-Register ein. Ansonsten wird der File-Eintrag nach dirEntryBuf ($8400) kopiert.
In R1 findet man Spur und Sektor des Directory-Blocks, in dem der gesuchte File-Eintrag steht. R5 speichert den Zeiger für den Directory-Eintrag in diskBlkBuf ($8000), dirEntryBuf ($8400) enthält den Pointer auf den Eintrag der gesuchten Datei im Inhaltsverzeichnis.

DeleteFile ($C238)

... tilgt die gewünschte Datei inkl. Directory-Eintrag (CBM-File-Typ = 0). Im Systemregister R0 muß die Adresse des File-Namens vermerkt sein, der im RAM steht. Wie gewohnt, speichert das x-Register die aktuelle Fehlerkennung (z.B. 5 = Datei nicht gefunden).

GetFile ($C208)

... ist die universelle Lade-Routine von Geos: alle File-Typen werden akzeptiert (auch z.B. Druckertreiber). Bei VLIR-Dateien läßt sich allerdings nur der erste Datensatz lesen.

Parameterangaben:
R6: Zeiger auf den Text des File-Namens im RAM.
Je nach File-Typ sind weitere Parameter zu übergeben (s. Tabelle).
Im x-Register wird stets die Fehlernummer vermerkt: aus Applikationen kann man nur bei Diskettenfehlern zur aufrufenden Routine zurückkehren; DeskAccessories springen auf jeden Fall zurück. Das gilt auch, wenn ein Daten-File (z.B. geladen wurde.

GetDirHead ($C247)

... benutzt die Routine GetBlock, um die BAM der aktuellen Disk im Laufwerk zu lesen und in curDirHead ($8200) abzulegen. 1571- und 1581-Blockbelegungspläne werden in dir2Head ($8900) gespeichert.
Im x-Register findet man die Fehlernummer, R4 zeigt auf curDirHead.

RstrAppl ($C23E)

... wird vom jedem DeskAccessory aufgerufen, damit es zu der Applikation zurückkehren kann, die es aktiviert hat. Intern lädt Geos das SwapFile und restauriert den Speicher. Laufwerkswechsel sind allerdings rückgängig zu machen - wenn sich das SwapFile nicht laden läßt, stürzt Geos nämlich ab.
Vorsicht: RstrAppl darf man niemals direkt mit dem Assembler-Befehl "jmp" aufrufen! Das klappt nur über einen kleinen Umweg: die Adresse von RstrAppl ist im Vektor "appMain" ($849B) einzutragen:

```
LoadW appMain,RstrAppl
rts
```

ChkDkGEOS ($C1DE)

... prüft, ob eine Geos-Disk im Laufwerk liegt. Trifft das zu, setzt Geos das entsprechende Flag (isGEOS = $848B). Normalerweise übernimmt das aber automatisch die Routine OpenDisk. Wer die Funktion trotzdem selbst programmieren will, muß vorher die BAM per Routine GetDirHead in den Speicher holen.

BlkAlloc ($C1FC)

... generiert eine Liste von Diskettensektoren und kennzeichnet sie in der BAM als belegt. Man übergibt der Routine eine bestimmte Byte-Anzahl (maximal 32258 = 127 Blocks auf Disk). Dann wird auf Diskette die jeweilige Anzahl der Sektoren in der BAM eingerichtet und die Track/Sektor-Tabelle angelegt. Das erste Byte des letzten Eintrags muß den Wert $00 als Endekennzeichen erhalten.

Parameterangaben:
R2: Anzahl der Bytes,
R6: Zeiger auf die Adresse im RAM, an der die Track-/Sektortabelle abgelegt wird.
Nach Durchlauf der Routine findet man in R2 die Anzahl der belegten Sektoren, in R3 Track- und Sektornummer des zuletzt generierten Diskettenblocks und im x-Register die Fehlernummer.

SaveFile ($C1ED)

... richtet normale Files ein und sichert gewünschte Speicherbereiche sowie leere VLIR-Files auf Disk.

Parameterangaben:
R10L: Directory-Seite, ab der man den nächsten freien File-Eintrag suchen will (die erste Seite ist stets Nr. 0).
R9: Zeiger auf den Info-Block des zu generierenden Files. Die Angabe der File-Struktur ist dafür verantwortlich, ob eine leere VLIR-Datei bzw. eine sequentielle erzeugt oder ein definierter Speicherbereich auf Disk abgelegt werden soll. Relevant ist Byte 70 des Info-Blocks: $00 = sequentiell, $01 = VLIR-Datei (Ladeadresse ist identisch mit Endadresse).
In beiden Fällen müssen die ersten Bytes des Info-Blocks den Zeiger auf den File-Namen der zu erzeugenden Datei enthalten ($00 als Ende-Kennzeichen). Beim Speichern ersetzt SaveFile diesen 16-Bit-Wert automatisch mit $00FF.
Nach dem Routinenablauf steht in R6 der Zeiger auf die Track-/Sektortabelle des generierten Files, in R9 findet man den Zeiger auf den Info-Block.

Parameterübergabe an GetFile

Geos-File-Typ	Systemregister	Funktion
alle Dateitypen	R6	Zeiger auf File-Name ($00 als Endekennzeichen)
Applikationen	R0L Bit 0 = 0	Programm an Default-Ladeadresse holen und sofort starten
	= 1	... an die in R7 angebene Adresse laden, aber nicht starten
	Bit 6 = 0	Daten-File (z.B. GeoWrite-Text) nicht drucken
		= 1 ... drucken
	Bit 7 = 0	Applikation muß kein Daten-File laden
	= 1	Daten-File laden. R3 enthält den Zeiger auf den Dateinamen, R2 den Pointer auf den Diskettennamen.
	R2	Zeiger auf Name der Daten-Disk
	R3	... auf den Namen des Daten-Files
DeskAccessories	R10L	muß auf $00 gesetzt werden

NewDisk ($C1E1)

... schickt den DOS-Befehl "Initialize" ans aktuelle Laufwerk. Die Routine wird z.B. von OpenDisk automatisch aktiviert. Parameter können bei dieser Routine entfallen.

RenameFile ($C259)

... erfüllt die gleiche Funktion wie die bekannte DOS-Anweisung des normalen C-64-Betriebssystems.

Parameterangaben:
R6: Zeiger auf alten File-Namen,
R0: ... registiert den neuen.
Achtung: Beide Dateinamen dürfen nicht im Bereich von diskBlkbuf ($8000) stehen!

FindBAMBit ($C2AD)

... gibt Hinweise über den Zustand eines Diskettensektors in der BAM. Dazu trägt man im Systemregister R6 die entsprechende Spur- und Sektornummer ein und erhält im Zero-Flag die gewünschte Information (0 = Block frei, 1 = Sektor belegt). Unser Listing zeigt ein entsprechendes Beispiel (Quelltext im GeoAssembler-Format). Nach dem Aufruf der Routine wird per BNE verzweigt, wenn der Block frei ist – BEQ zeigt an, daß er belegt ist.
Die folgenden Register sind nur bei einem 1541-Treiber relevant:
x-Reg.: Index auf BAM-Byte,
R8H: Bit isolieren (AND R8H),
R7H: Index auf freie Blocks in BAM

(CalcBlksFree ($C1DB)

... berechnet die Gesamtzahl der freien Blöcke auf Disk (per Gesamtzahl in der BAM) und gibt den Wert im High-Low-Byte-Format aus. Die Routine wird von den meisten anderen Kernel-Programmen automatisch eingesetzt und ist für Programmierer relativ bedeutungslos.

Parameterangaben:
R5: Position der BAM-Einträge im RAM (normalerweise curDirHead).
Nach Aufruf steht im x-Register die Fehlernummer, in R4 die Anzahl der freien Blöcke und in R3 zusätzlich die Gesamtzahl aller verfügbaren Blöcke auf Disk (664 bzw. 1328).
In R4 erhält man die Anzahl der freien Blöcke auf Diskette.

PutDirHead ($C24A)

... holt den Inhalt der BAM aus curDirHead ($8200) bzw. dir2Head ($8900) und schreibt die Bytes auf die aktuelle Disk im Laufwerk. Die Routine ist stets aufzurufen, wenn man den Blockbelegungsplan manipuliert (z.B. mit AllocateBlock).

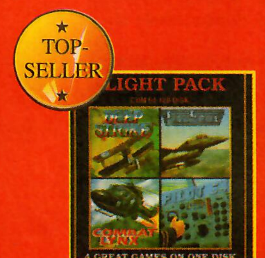
Listing. Beispiel zu FindBAMBit

```
Spur = 18
Sektor = 10
xKoord = 500 ;für Ausgabe per
yKoord = 100 ;PutString
BlockDef:
LoadB r6L,#Spur
LoadB r6H,#Sektor
jsr FindBAMBit
beq Ausgabe ;Zero-Flag-Abfrage
LoadW r11,#xKoord
LoadB r1H,#yKoord
LoadW r0,not
jmp PutString
Ausgabe:
LoadW r11,#xKoord
LoadB r1H,#yKoord
LoadW r0,yes
jmp PutString
yes:
.byte "Dieser Sektor ist belegt.",NULL
not:
.byte "Dieser Block ist frei.",NUll
```

ORIGINAL-SOFTWARE

Endlich gibt es sie wieder! Die besten Spiele für den C 64. Direkt

Greifen Sie zu, solange der Vorrat reicht! (Spiele nur noch in begrenzter Menge vorhanden!)



Flight Pack

Mit dem "Flight Pack" haben Sie vier mal die Chance Herr der Lüfte zu sein. Neben "Deep Strike" und "Combat Lynx", auch die knackige Bombersimulation "First Strike".
Best.Nr. 641119
nur DM 9,80



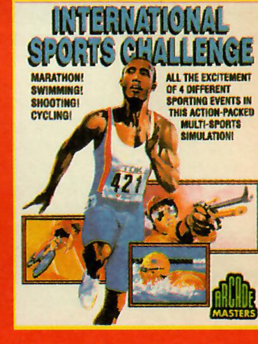
Heavenbound

T-Fant wird gesucht! Begeben Sie sich zu einem heißen Jump'n Run-Abenteuer als Elefanten-Held in Richtung Himmel.
Bestell-Nr. 649410
nur DM 19,80



Motley Tetris

Die Variante des Tüftel-Hit aus Russland bunt wie noch nie und garantiert suchterzeugend
Bestell-Nr. 649409
nur DM 9,80



International Sports Challenge

Eine aktionsgeladene Multi-Sport-Simulation für einen oder zwei Spieler: Marathon, Schwimmen, Schießen und Radfahren!
Best.Nr. 649417

Preissensation: DM 9,80

BESTELLCOUPON

Ich möchte folgende Software bestellen:

	Anzahl	Softwarebezeichnung	Bestell-Nr.
1.			
2.			
3.			
4.			
5.			
6.			
7.			
8.			

ABSENDER (Bitte leserlich ausfüllen)

Name, Vorname

Straße / Nr.

PLZ Ort

Bitte ausschneiden und absenden an:
ERDEM Development,
Postfach 1823, 84471 Waldkraiburg oder
Tel. 08638 / 9670-70, Fax 08638 / 9670-55

Gewünschte Zahlungsweise bitte ankreuzen:
Ausland nur gegen Vorkasse mit Euroscheck oder Postanweisung; zzgl. DM 12,- (Versand, Porto)

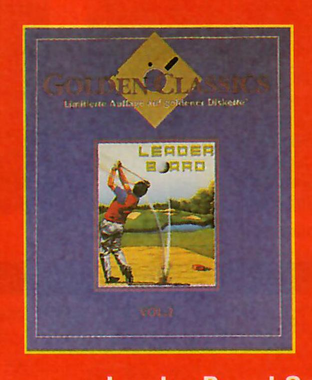
- ☐ Vorkasse mit V-Scheck (Versandkostenpauschale 7,- DM)
- ☐ Per Nachnahme (Versandkostenpauschale 12,- DM)
- ☐ Bankabbuchung (Versandkostenpauschale 7,- DM)

Bankleitzahl

Konto-Nr. Kontoinhaber

Geldinstitut

Datum, Unterschrift (bei Minderjährigen des gesetzl.Vertreters)



Leader-Board Golf

Spielen Sie wie Bernhard Langer oder Nick Faldo auf Ihrem C 64.
Best.Nr. 640702

Preissensation: DM 9,80

ZUM KNÜLLERPREIS!

von uns. So billig war Original-Software noch nie! Packen Sie zu!

TOP-SELLER



Big Box

30 Spiele in einer Box voller Action und Strategie. Darunter viele Klassiker wie Wonderboy, Ghostbuster, Spindizzy, Rampage und Hacker.
Best.Nr. 640601
nur DM 29,90

AUSVERKAUFT



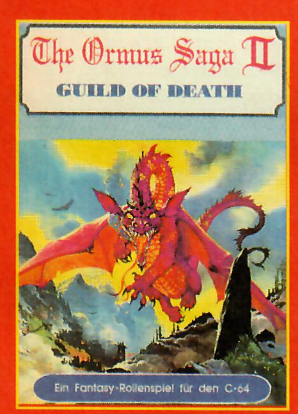
Big Box 2

Phantastische Spiele-Sammlung mit dreißig Titeln: u.a. "Driller", "Hunters Moon", "Gribbly's Day Out" und "Sanxion
Best.Nr. 649401
nur DM 29,90



Sword of Honour

Finden Sie als Ninja das Schwert des Shoguns! Retten SIe seine Ehre und bewahren SIe ihn vor dem Freitot durch Harakiri. Zwei Disketten voll mit Action und vielen Rätseln. Fantastische Grafiken entführen Sie in das ferne Japan.
Best.Nr. 649418
nur DM 29,80



The Ormus Saga II

Kämpfe gegen Drachen, Dämonen, Zauberer und andere Bösewichte und beweise taktisches Geschick in der Schlacht gegen die Truppen des Ormus-Kultes.
Best.Nr. 649404
nur DM 24,90



Soccer Mania

Die Fußballspiel-Sammlung der Extra-Klasse mit "Football Manager 1 und 2 " und dem Ultra-Hit "Microprose Soccer"
Best.Nr. 649402,
nur DM 19,90

NEU!



"Spiele-Koffer"

Drei heiße Games der Spitzenklasse auf einen Schlag: Zak McKraken, Oil Imperium und European Soccer. Der Preisknüller und Geheimtip für Spieler!
Best.Nr. 649411

Preissensation: DM 29,80

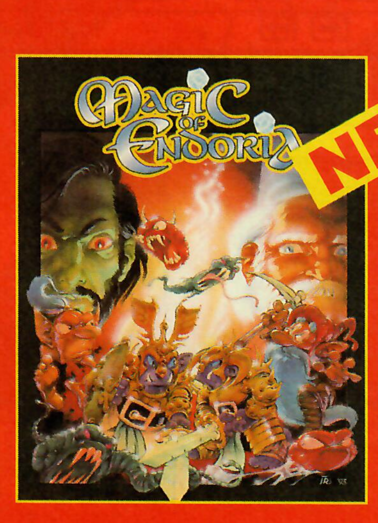
NEU!



Ormus Saga III

Der abschließende Teil der Ormus-Saga-Trilogie entführt den Spieler erneut nach Beryland, wo es gilt, zahlreiche Abenteuer zu bestehen. Auch für Neueinsteiger interessant, da man keinen der beiden Vorgänger kennen muß!
Best.-Nr. 649407
Preis: 39,90 Mark

NEU!



Magic of Endoria

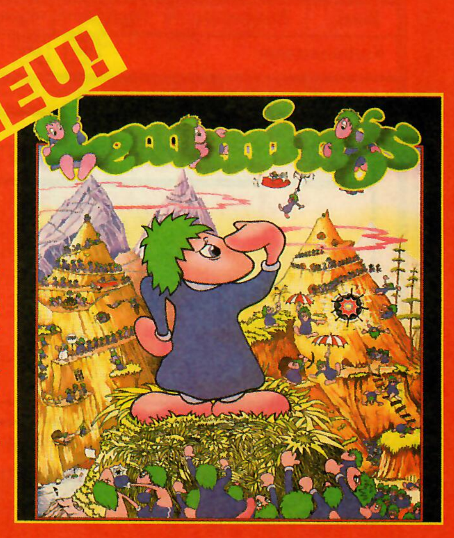
Graben Sie im Erdreich von Endoria nach den geheimnisvollen Kristallsplittern und steigen Sie zum Gott auf. Eine Mischung aus Fantasy und Handelssimulation.
Best.-Nr. 649406
Preis: 49,90 Mark

NEU!

Lemmings

Die dümmlichen Nager sind da und verbreiten auf fünf Disketten Spielefieber und Suchterscheinungen. Bringen sie die kleinen Kerle mit der richtigen Strategie heil durch die 120 Level.
Best.Nr. 649405

Preissensation: DM 29,80



Erste Schritte mit GeoProgrammer



Geos zum Anfassen

Das beste Entwicklungspaket für Geos-Applikationen (MegaAssembler) ist vom Markt verschwunden – GeoProgrammer schließt die Lücke. In der neuen Folge unseres Kurses wollen wir die Zählroutine für die Pfeil-Gadgets in der Menüleiste für die Kartenanzeige in den GeoWrite-Quelltext integrieren.

Wir haben es zwar in der letzten Folge unseres Geo-Programmer-Kurses nicht ausdrücklich erwähnt, aber beim Durchforsten des GeoWrite-Quelltextes zur Applikation "CardBox" auf Diskette sind Sie bestimmt schon selbst daraufgekommen: Selbstverständlich müssen die Funktionen des Labels "Icon-Handler" im Hauptprogrammteil aktiviert werden – das geht mit einer zusätzlichen Anweisungszeile. Fügen Sie im Label "Menu-Prg" (GeoWrite-Quelltext, Seite 2) hinter "jsr DoMenu" ein:

```
jsr IconHandler
```

Alles weitere erledigt die damit aufgerufene Geos-Kernel-Routine **DoIcons**.

Zählwerk für Datensätze

Erinnern Sie sich an die beiden Pfeil-Icons, die wir in der letzten Kursfolge als Grafik-Images in unsere Applikation eingebaut haben? Sinn und Zweck dieser Piktogramme ist, den vorhergehenden bzw. folgenden Datensatz per Mausklick zu holen und auf den Bildschirm zu bringen. Deshalb sollten wir schleunigst die entsprechende Zählroutine entwerfen:

❑ Pfeil nach links: Datensatznummer reduzieren,

❑ Pfeil rechts: ... erhöhen.

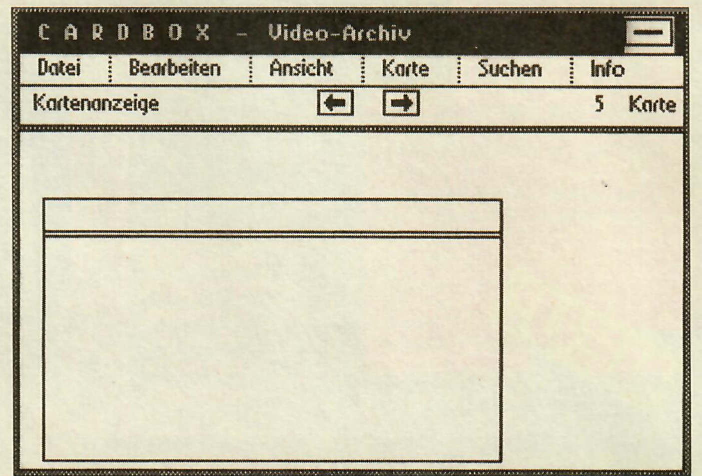
Wertvolle Dienste für dieses Vorhaben leistet die bereits integrierte Routine "Counter", die nach Programmstart den Ausgangswert in die Menüleiste "Kartenanzeige" auf den Screen schreibt: 0 Karte. Dieses Unterprogramm ist lediglich entsprechend zu manipulieren, um den jeweils aktuellen Wert anzuzeigen. Und das geht am besten, wenn wir im Label "IconHandler" (Listing s. 64'er 6/95) die beiden Aktions-Routinen der Suchpfeile ändern.

Löschen Sie also hinter "AcIc-Left" und "AcIcRight" die Dummy-Anweisung "jmp ProgStart" und ergänzen Sie den Quelltext mit den Befehlen lt. unserem nebenstehenden Listing.

Das Programm greift auf den Wert zurück, der im Platzhalter "Record" steht. Dabei lernen wir die Funktion zweier GeoProgrammer-Makro-Befehle kennen, die in der Systemdatei "geosMac" enthalten sind:

❑ **CmpWI:** ... vergleicht den Inhalt einer 16-Bit-Adresse mit einem absoluten Wert. Bei der Routine für den Linkspfeil wird überprüft, ob die Zahl 0 in "Record" steht. Trifft es zu, verläßt "CardBox" das Unterprogramm (s. Label "IconR11")- sonst würden falsche Werte entstehen (Sie wissen, daß die Reduzierung des Wertes 0 bei allen Assembler-Programmen nicht "-1", sondern "255" ergibt!). Bei jeder anderen Zahl in "Record" wird ein Zähler abgezogen und der neue Wert per Unterprogramm "Counter" auf dem Bildschirm ausgegeben.

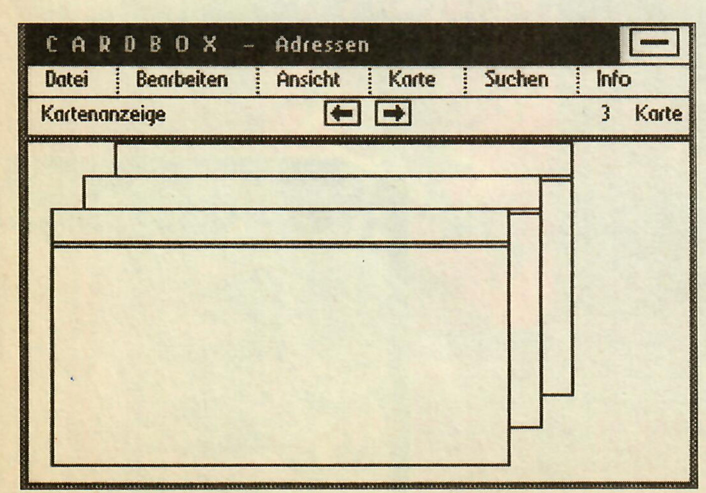
❑ **CmpW:** ... überprüft die Inhalte (16-Bit-Zahl) zweier durch Label-Namen definierten Speicherstellen auf identische Werte (hier gibt es also keine absolute Vergleichszahl wie bei "Cmp-WI"!). Ansonsten ähnelt die Zählroutine für den Rechtspfeil der anderen aufs Haar - allerdings wird die aktuelle Zahl in "Record" um "1" erhöht. Auch hier ist ein Sicherheitsfaktor eingebaut: Im Geos-Betriebssystem existiert eine spezielle Adresse ($8497), die sich bei allen Dateiverwaltungs-Applikationen um den reellen Wert der tatsächlich vorhandenen Datensätze kümmert. Deren symbolische Bezeichnung ist in der GeoProgrammer-Datei "geos-Sym" enthalten: **usedRecords**. Übersteigt also die vom Programm manipulierte Zahl in "Record" den echten Wert, macht Geos nicht mit und verläßt die Zählroutine (s. Label "IconR21"). Nur, wenn die in "Record" verankerte Zahl geringer ist als "used-Records", wird sie um einen Zähler erhöht.



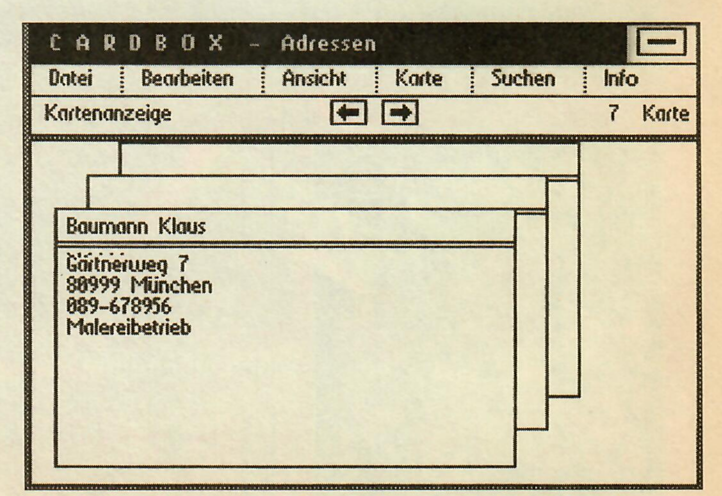
Die eingebaute Zählroutine klappt: per Linkspfeil blättert man zurück, der Rechtspfeil holt den nächsten Datensatz

Datensätze eingeben

Die soeben entworfene Zählroutine ist im GeoWrite-Quelltext "CardBox.KURS" bereits enthalten. Laden Sie also die daraus entstandene Applikation "CardBox" von der Programmservice-Disk und probieren Sie das neu integrierte Zählwerk aus (Mausklick auf beide Pfeil-Icons).



Optimierter Eingabe-Bildschirm mit drei Karteikarten (Ergänzung des Quelltextes im nächsten Heft)



Die Eingabe-Routine fürs Kartenfeld greift aufs Geos-Kernel-Unterprogramm GETSTRING zurück

Endlich: jetzt wollen wir die Programmierung der Hauptfunktion unserer Geos-Applikation in Angriff nehmen – die Eingabe und Sicherung der Datensätze in einer entsprechenden VLIR-Datei ("CardBox Data").

Zunächst wollen wir uns überlegen, durch welchen Menüpunkt die Dateneingabe aktiviert werden soll. Halten wir uns auch hier ans große Vorbild (Applikation "Kartei" in der PC-Benutzeroberfläche WINDOWS): dort lassen sich neue Datensätze eingeben, wenn man den Menüpunkt "Hinzufügen" (Pulldown-Menü "Karte") aufruft.

Beim Durchforsten unseres GeoWrite-Quelltextes werden wir schnell fündig (Seite 4): der Label "AcHinzu" muß neu programmiert werden (löschen Sie also die Anweisung "jsr ReDoMenu"). Das ist nicht schwierig, da uns das bereits bestehende Unterprogramm "AddBorder" bei unserem Vorhaben unterstützen soll.

Neu zu schaffen sind allerdings Routinen, die den alten Namen der Karteikarte (Stichwort) löschen, einen Datenpuffer für die Stichworteingabe sowie eine spezielle Eingabe-Box zur Verfügung stellen. Last but not least ist zu testen, wieviele Zeilen pro Datensatz (Datenfelder) auf den Eingabe-Screen passen und wie breit jedes Datenfeld maximal sein darf. Auch hier bleibt´s uns nicht erspart: pro Datenfeld ist im RAM zunächst entsprechender Speicher zu reservieren (zumindest bis der Datensatz in der vorgesehenen VLIR-Datei gespeichert wird).

Sparen Sie sich die Mühe: wir haben´s bereits ausgerechnet – auf der Screen-Karteikarte unserer Applikation haben (ohne Stichwort) exakt zehn Zeilen Platz, die jeweils nicht länger als 43 Bytes sein dürfen (= 473 Bytes inkl. Stichwort).

Ergänzen Sie den GeoWrite-Quelltext unseres Programmprojekts mit den entsprechenden Labels und Tabellen (s. Listing). Die neuen Programmzeilen dürfen Sie getrost ans bisherigeTextende anfügen.

Weniger interessant ist die Reservierung der Datenpuffer für die jeweiligen Eingabezeilen der Karteikarte (das erledigt prompt und zuverlässig die inzwischen bekannte GeoProgrammer-Direktive ".block"). Die Programmzeilen im Label "HinzuBox" sind dagegen ein Paradebeispiel für die Umsetzung der Geos-Kernel-Routine "DoDlgBox" in die Praxis (näheres s. "Geos intern", 64´er 2/95). Zunächst bestimmt man Breite und Höhe des Bildschirmfensters und läßt dasRechteck auf dem Screen erscheinen (DBGRPHSTR, unterstützt von den bereits definierten Parametern zu "AddBorder"). Dann folgt der Parameterblock zur DoDlgBox-Funktion DBGETSTRING (Texteingabe, maximal 42 Zeichen – das 43. Null-Byte muß als Endekennung herhalten. Zum Schluß wird die Eingabe-Box noch mit dem OK-Icon ausgestattet. Die Routine im label "DelCdNme" löscht den alten Inhalt des Eingabepuffers für den Namen der jeweiligen Karteikarte (Stichwort) und verwendet dazu "ClearRam" ($C178).

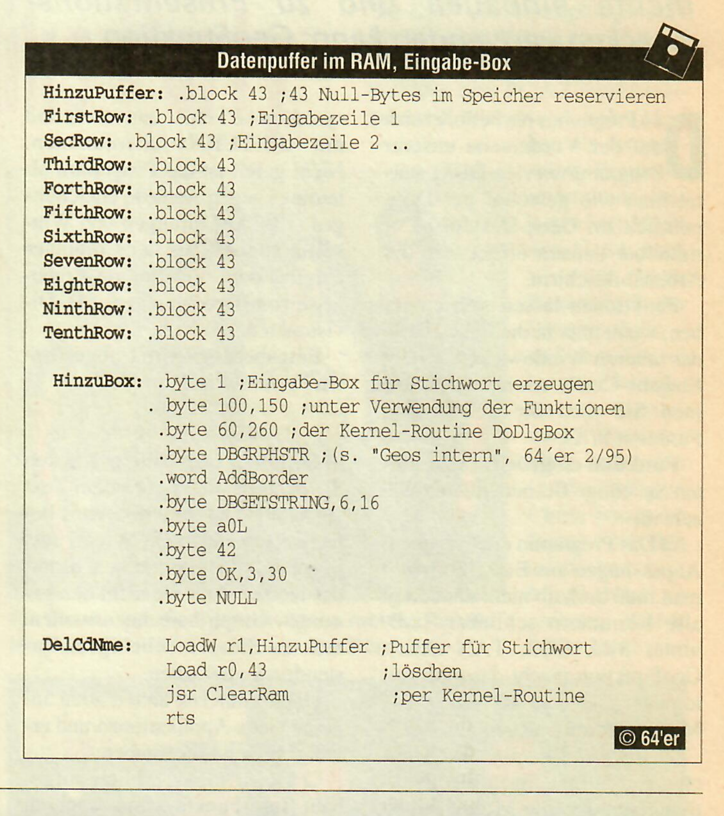
Harald Beiler

Labels "AcIcLeft"/"AcIcRight" (Links- und Rechtspfeil)

```
............ (Quelltext wie gehabt)
AcIcLeft:
            CmpWI Record,0   ;Nr. 0 im Label "Record"?
            beq IconR11      ;ja, zurück zur MainLoop
            dec Record       ;Wert reduzieren
            jsr Counter      ;Bildschirmausgabe
IconR11:    rts

............ (Quelltext wie gehabt)

AcIcRight:
            CmpW Record, usedRecords ;aktuelle Datensatzanzahl?
            beq IconR21              ;ja, zurück zur MainLoop
            inc Record               ;Wert erhöhen
            jsr Counter              ;Bildschirmausgabe
IconR21:    rts
```

Datenpuffer im RAM, Eingabe-Box

```
HinzuPuffer: .block 43 ;43 Null-Bytes im Speicher reservieren
FirstRow: .block 43 ;Eingabezeile 1
SecRow: .block 43 ;Eingabezeile 2 ..
ThirdRow: .block 43
ForthRow: .block 43
FifthRow: .block 43
SixthRow: .block 43
SevenRow: .block 43
EightRow: .block 43
NinthRow: .block 43
TenthRow: .block 43

 HinzuBox: .byte 1 ;Eingabe-Box für Stichwort erzeugen
          . byte 100,150 ;unter Verwendung der Funktionen
           .byte 60,260 ;der Kernel-Routine DoDlgBox
           .byte DBGRPHSTR ;(s. "Geos intern", 64´er 2/95)
           .word AddBorder
           .byte DBGETSTRING,6,16
           .byte a0L
           .byte 42
           .byte OK,3,30
           .byte NULL

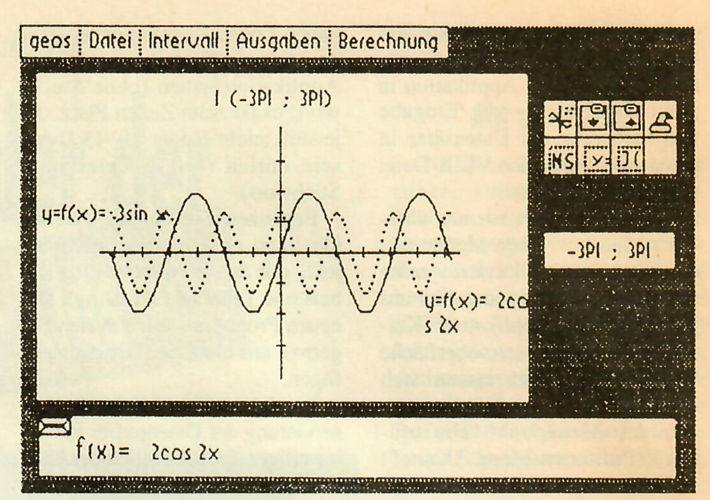
DelCdNme:   LoadW r1,HinzuPuffer ;Puffer für Stichwort
            Load r0,43           ;löschen
            jsr ClearRam         ;per Kernel-Routine
            rts
```


Dieses Symbol zeigt an, welche Programme auf Diskette erhältlich sind

Geos-Programme auf Disk

Neues von Geos

Neben den üblichen Dateien zu unserem Geo-Programmerkurs bieten wir Ihnen heute auf unserer Programmservice-Disk eine komfortable Applikation zum Zeichnen beliebiger Funktionsgraphen, die man in eigene Dokumente einbauen und zu Präsentationszwecken verwenden kann: GeoFunktion.



Der fertige Funktionen-Graph im Grafik-Window von GeoFunktion. Die entsprechenden Intervalle erscheinen rechts in der Mitte.

Das Programm (es befindet sich auf der Vorderseite unserer Programmservice-Disk) startet man wie gewohnt per Doppelklick im Geos-Desktop. Unmittelbar danach öffnet sich der Arbeitsbildschirm.

Funktionen lassen sich eingeben, wenn man in die linke Hälfte des unteren Windows klickt – der Eingabe-Cursor erscheint. Bestimmen Sie jetzt die gewünschte Funktion F(X).

Funktion eingeben: Dazu sollten Sie einige Besonderheiten beachten:

❑ Das Programm erledigt intern Anpassungen ans Basic-Format – man muß deshalb nicht unbedingt alle Klammern schließen (z.B. hinter SIN, COS, TAN usw.): GeoFunktion macht das ganz automatisch, ebenso das Setzen von Multiplikationszeichen.

❑ Bei der Eingabe von Gleichungen nimmt Ihnen GeoFunktion ebenfalls jede Menge Arbeit ab – Funktionsbezeichnungen lassen sich mit einem Buchstaben abkürzen: S = SIN, C = COS, T = TAN, A = ATN, W = SQR (Wurzel), L = LOG und P = pi.

❑ Potenzen sind per Tastenkombination <SHIFT Pfund> anzugeben – jetzt erscheint der Cursor nach oben versetzt und erlaubt die Eingabe der gewünschten Potenzzahl. Zurück zur normalen Ebene geht´s per <SHIFT +>.

❑ Die Rechensymbole entsprechen den gewohnten Tasten (also z.B. "/" für Division), außerdem gilt "e" für die Eulersche Zahl (2.7182818285).

❑ Alle Zeichen kann man wie gewohnt mit der DELETE-Taste <INST/DEL> löschen.

Drückt man während der Eingabe <RUN/STOP>, löscht diese Aktion die neu definierte Gleichung und bringt wieder die alte Formel. Die Eingaben werden auf syntaktische Fehler untersucht und mit <RETURN> übernommen. Nicht gefeit ist das Programm allerdings gegen sinnlose Gleichungen (z.B. Anhäufungen von Winkelfunktionen, bis der Computer aufgibt) oder undefinierte Ergebnisse von Berechnungen, z.B. Divisionen durch Null.

Eingabebeispiel mit abgekürzten Formelwerten:

```
3sin 3x = 3*sin (3x)
3.9x/(4+x = 3.9*x/(4+x)
```

Graphen berechnen: Klicken Sie dazu das Icon über dem Text "F(X) =". Die Funktion wird berechnet und auf dem Screen ausgegeben (währenddessen blinkt das Icon). Wenn Sie nicht den gesamten Graph berechnen wollen, müssen Sie per beliebigem Tastendruck abbrechen.

Die Menüleiste ähnelt allen anderen Geos-Applikationen und erledigt folgende Aufgaben:

❑ Geos: ... wie üblich aufgebaut (Info-Funktion und eventuell vorhandene DeskAccessories).

❑ Datei: öffnet ein Pulldown-Menü mit folgenden Untermenüpunkten:

– *drucken:* ... schickt den Inhalt des Grafikfensters zum Drucker (bauen Sie zur Sicherheit das Window vorher nochmals auf, s. Erläuterung zu "Icons"). Selbstverständlich muß sich auch der für Ihren Drucker relevante Treiber auf der Disk bzw. im gewählten Laufwerk befinden (z.B. geoRam, REU 1764).

– *Ende:* Rückkehr zum Desktop, Programmende.

❑ Intervall: ... läßt sich mit diesem Menüpunkt bestimmen. Zehn verschiedene Funktionsintervalle stehen zur Verfügung (Minus- und Pluswerte als Bereichsgrenzen):

normal: (-2; 2), (-4; 4), (-6; 6), (-8; 8), (-10; 10),

trigonom: (-pi; pi), (-2pi; 2pi), (-3pi; 3pi), (-4pi; 4pi), (-5pi; 5pi).

❑ Ausgaben: ... macht ebenfalls ein Pulldown-Menü auf:

– *Punktmode*: zeigt sich dieser Menüpunkt kursiv, werden alle nicht direkt nebeneinanderliegende Punkte durch Linien miteinander verbunden,

– *Nullstellen/Gleichung/Intervall/Photoscrap*: Damit legt man fest, welche der genannten Optionen im Grafik-Window erscheinen soll. Im Pulldown-Menü inaktive Punkte erscheinen kursiv.

Nullstellen beliebig verschieben

❑ Berechnung: Das sind die einzelnenUntermenü-Punkte:

– *Nullstellen:* ... berechnet sie im gewählten Intervall. Die Gleichung muß bereits eingegeben und das Intervall gewählt sein. Die entsprechende Anzeige gibt darüber Auskunft, wie weit der Computer mit seinen Berechnungen ist (mit jeder beliebigen Taste kann man jederzeit abbrechen). Wenn´s mehr als 10 Nullstellen sind, macht das Programm Schluß und gibt die bislang berechneten Nullstellen aus – natürlich auch dann, wenn die Berechnung ganz normal beendet wurde. Nach Tipp auf die Maustaste erscheint eine bewegliche Box, die man innerhalb des Grafikfensters beliebig positionieren kann (fixieren ebenfalls per Mausklick) – dort erscheinen dann die Nullstellen. Möchten Sie auf die Anzeige verzichten, ist die Box über dem Grafikfenster zu positionieren (Bestätigung per Mausklick). Will man die Nullstellen-Anzeige neu positionieren, geht das mit dem entsprechenden Icon (s. Erläuterung).

– *Einzelwerte:* ... berechnet gewünschte Werte einer Gleichung, die Wahl eines Intervalls ist dazu nicht notwendig. Eingaben erledigt man im üblichen Basic-Format. Das Programm checkt die Eingaben auf Fehler und bringt eine entsprechende Meldung, wenn es fündig wird. Will man die Eingabe beenden, genügt der Tipp auf <RETURN>, ohne irgendwelche Zahlenangaben.

Icon-Bar: Auf dem Bildschirm oben rechts finden Sie sieben Icons (das achte ist ohne Funktion), die von links nach rechts folgende Aufgaben erfüllen:

1. ... zeichnet die gesamte Grafik neu. Wenn der Graph schon vorher berechnet wurde, nutzt das Programm die fixierten Punkte aus – das spart immens Rechenzeit. Alle bereits positionierten und aktivierten Daten (z.B. Gleichung) werden erneut ausgegeben.

2. ... dient zum Einkleben eines Photoscraps. Es läßt sich aber nicht frei positionieren, da seine Hauptfunktion die gleichzeitige Anzeige mehrerer Graphen ist.

3. .... sichert das Bild als Photoscrap auf Diskette. Ab sofort kann es wie gewohnt in andere Geos-Applikationen (GeoWrite, GeoPaint, GeoPublish) eingebaut werden. Es dient zur grafischen Gestaltung solcher Dokumente.

4. ...aktiviert die Druckausgabe des Grafikfensters (funktioniert wie Menüpunkt "Datei/drucken").

5. ... positioniert die Nullstellen: Nach dem Klicks aufs Icon wird der Mauszeiger ins Grafik-Window transferiert, die Nullstellen lassen sich nun an beliebig anderer Stelle einrichten. Hat man die Nullstellen-Anzeige durch den entsprechenden Menüpunkt unterbunden, werden die Ziffern erneut ausgegeben und per Mausklick neu positioniert.

Die restlichen beiden Icons ("Funktionstext und Nullstellen positionieren") funktionieren nach Mausklick analog dazu.

Andre Marth/bl

Command Line Interpeter CLI 3.0

Geos per Tastatur

Nicht immer sind Eingaben per Maus oder Joystick der Weisheit letzter Schluß. Mit der Befehlseingabetechnik von CLI 3.0 kommt MS-DOS-Feeling auf!

Vor mehr als einem Jahr kam der Geos-Command-Line-Interpreter in Version 2.0 erstmals auf den Markt. Die neueste Fassung läßt sich getrost als endgültig bezeichnen: jede Menge neuer Befehle und Funktionen sind dazugekommen, die das Arbeiten mit dieser MS-DOS-ähnlichen Benutzeroberfläche zum Vergnügen machen.

Das Programmpaket erscheint innerhalb der GUC-Reihe "Geos Professional" und besteht aus einer 5,25-Zoll-Disk inkl. 32seitigem Handbuch. Vor dem Programmstart muß man sich an die unvermeidliche Installation machen – CLI 3.0 läßt sich für Geos 64 und Geos 128 (allerdings nur im 80-Zeichenmodus) einrichten. Dazu ist nach Programmaufforderung eine spezielle Code-Bezeichnung lt. Tabelle im Handbuch anzugeben, die künftig in Ihrer individuellen CLI-Version integriert ist (quasi als Kopierschutz, dieses Methode verwenden auch diverse Spielehersteller). Beachten Sie, daß zusätzlich die Kennzahl Ihrer individuellen Geos-Version mitgespeichert wird (vergleichbar mit der Applikation GeoWrite). Die CLI-Systemdisk wird bei der Installation durch wichtige Dateien ergänzt: Key-Manager, InitCLI (letztere braucht man unbedingt, um die neue Benutzeroberfläche zu initialisieren und zu starten).

CLI
V 3.0
Command Line Interpreter
©1994 by Jürgen Eckel
Willkommen zum CLI-Installationsprogramm !
Installation OK !
DeskTop
Install128

Installation muß sein! Auf der Systemdisk werden zusätzliche Geos-Files eingetragen, die man zum Programmstart braucht.

Ein Hauch von MS-DOS

CLI verzichtet gänzlich auf die grafischen Elemente des Geos-Desktop und stellt lediglich einen Eingabe-Prompt (Buchstabe des jeweils aktivierten Laufwerks, z.B. A fürs Floppy-Drive oder B für die RAM-Disk) in der aktuellen Befehlszeile zur Verfügung. Der Vorteil: Anweisungen und Befehle sind oft schneller eingetippt als per Mauszeiger innerhalb unzähliger Pulldown-Menüs aktiviert. Bei dem relativ geringen Speicherbedarf (20 KByte) steht CLI 3.0 komplett im Rechnerspeicher – es werden also keine Programmteile zeitraubend von Diskette nachgeladen: alle Funktionen stehen unmittelbar nach Befehlseingabe zur Verfügung. Bemerkbar macht sich das vor allem bei Kopier-Aktionen: man hat deutlich mehr RAM zur Verfügung.

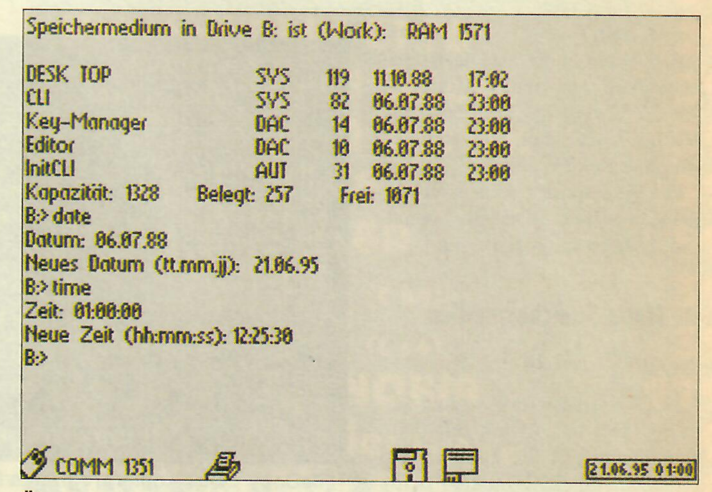
Wer sich mit IBM-kompatiblen PCs auskennt, wird mit den CLI-Anweisungen kaum Schwierigkeiten haben, z.B. liefert CHKDSK nützliche Infos zum aktuellen Laufwerk (welcher Floppytyp, freie Blocks auf Disk usw.). DATE und TIME stellen Datum und Zeit schneller ein als beispielsweise die Manipulation der entsprechenden Fenster im Desktop rechts oben. Und DIR bringt alle File-Einträge der Diskette im aktiven Laufwerk. Insgesamt existieren 24 CLI-Befehle, die in der Schreibweise MS-DOS ähneln.

Wie die MS-DOS-Computer kennt CLI 3.0 ebenfalls jede Menge Tastaturfunktionen (Short-Cuts), die z.B. Textausgabe (aktiviert per TYPE) unterbrechen oder sämtliche CLI-Kommandos auf dem Bildschirm zeigen. Die vier C-64-Funktionstasten sind vierfach belegt und bringen auf Tastendruck voreingestellten Text bzw. CLI-Befehle, die sofort ausgeführt werden (z.B. F1 mit der Anweisung DIR/W). Diese Texte erzeugt man mit dem DeskAccesory "Key-Manager" im entsprechenden Editor-Bildschirm für die 16 Funktionstasten. Nicht vergessen: neue Einträge werden erst nach <RETURN> übernommen! Per KEY-Anweisung bringt man die Funktionstastenbelegung auf den Bildschirm.

Unverzichtbar für den Systemstart ist die neu installierte Datei "InitCLI". Damit lassen sich Paßwort, Systemschutz oder Terminüberwachung auf Wunsch einstellen – aber man kann auch festlegen (per Klick aufs entsprechende Icon), ob CLI nach jedem Booten des Geos-Systems die Herrschaft übernehmen soll (statt DeskTop bzw. TopDesk). Dazu ist lediglich die gewählte Setup-Konfiguration auf Disk zu speichern – ab sofort gilt diese Einstellung bei jedem Geos-Start. Natürlich läßt sich diese Konfiguration auf Wunsch jederzeit wieder ändern (falls DeskTop beim Booten aktiviert werden soll).

Auf einen Blick

Sicher ist die Befehlszeilen-eingaben-Philosophie von CLI 3.0 nicht jedermanns Sache (die Menüführung per Mauszeiger mit Icons und Grafik-Windows hat jede Menge Fans), dennoch merkt man vor allem bei Disketten-Operationen schnell die immensen Vorteile von CLI: Laufwerke lassen sich per simpler Eingabe des Geräte-Buchstabens blitzschnell wechseln, Directories erscheinen im Handumdrehen auf dem Bildschirm. Schade, daß es den hilfreichen Editor zur Entwicklung von Batch-Dateien nur auf einer gesonderten Diskette gibt (neben weiteren CLI-Utilities, die man zusätzlich kaufen muß). *bl*



Ähnelt in jeder Beziehung der Befehlseingabe von MS-DOS: CLI 3.0 stellt 24 Anweisungen zur Verfügung

64'er-Wertung: Command Line Interpreter CLI 3.0

Alternative zur grafischen Benutzeroberfläche des Geos-Desktop nach dem Muster des PC/AT-Systems MSDOS

Positiv

- knappe Befehlseingabe
- übersichtliches Handbuch
- Programm resident im Speicher
- Funktionstastenbelegung läßt sich ändern

Negativ

- umständliche Installation
- Batch-Editor nur auf zusätzlicher Disk erhältlich

Wichtige Daten

Bezugsquelle: Geos User Club GbR, Moerser Str. 11, 46286 Dorsten, Tel. u. Fax: 02866-376
Preis: 25 Mark inkl. Anleitung
Testkonfiguration: C 128D, REU 1750, Floppy 1571/1581

Beurteilung:

SEHR GUT

System-Erweiterung

GoDot greift nach Mega-Bytes!

Egal mit welchem Computersystem man arbeitet, wenn Grafik ins Spiel kommt benötigt der User viel Speicherplatz. Dieser Tatsache tragen die Entwickler von GoDot jetzt Rechnung und haben den Image-Prozessor auf die speichermächtigen Datenträger von CMD und Co. angepaßt.



Bisher war der Zugriff auf standardisierte Commodore-Laufwerke unter GoDot kein Problem. Der Image-Prozessor erkannte beim Start automatisch alle angeschlossenen Devices. Mit den Partitionen einer Festplatte oder Floppy 1581 konnte das Programm aber nicht viel anfangen. Dieses Manko soll eine erweiterte GoDot-Version beseitigen.

Die Installation

Um die CMD-Geräte unter GoDot zu nutzen, müssen exakt fünf Files vom Original-Programm durch modifizierte Dateien ersetzt werden. Dazu gehören:

godot
god.main
dev.REU
mod..FileCopy
nod..REUTool

Außerdem findet man nach dem Entpacken ein neues Modul – *"mod..ChangeDir"*. Es dient zum Wechseln zwischen Verzeichnissen ohne Befehlseingabe.

Die Files werden einfach auf eine GoDot-System-Diskette kopiert und fertig ist die Installation.

Neue Speichermedien im Test

GoDot wird wie gewohnt gestartet. Bei der Arbeit mit dem Programm fällt die Einbindung der neuen Speichermedien beim ersten Hinsehen nicht auf.

Erst beim Speichern und Laden werden die CMD-Laufwerke angezeigt. Dabei greift das System ohne Murren auf alle bekannten CMD-Devices zu. Unser Test erfolgte mit einer Festplatte "HD 40", einem "RAM-Link" und einem Diskettenlaufwerk "FD-4000". Speichern, Laden und der Programmstart funktionierten mit diesen Medien ohne Probleme.



"FileCopy": Das Kopierprogramm für GoDot wurde nun auch an andere C-64-Speichermedien angepaßt

Neue Module unter der Lupe

Neben den verbesserten System-Files findet man im Programm-Paket zusätzliche zwei Programm-Module:

Mit *"mod.ChangeDir"* soll dem User die Arbeit mit den Unterverzeichnissen erleichtert werden. Mit dessen Hilfe ist die "Hangelei" durch Sub-Directories einer Festplatte oder Floppy recht einfach und erspart dem Anwender die umständliche Eingabe der entsprechenden Befehle. Leider sorgt ein kleiner Fehler bei der Nutzung für Probleme: Versucht man mit *"mod.ChangeDir"* auf einem Datenträger ohne Unterverzeichnisse ins Root-Directory zu springen, hängt sich der C 64 auf!

Das Modul *"mod..FileCopy"* ist das zweite Zusatzprogramm im Bunde. Bisher kommunizierte das Kopier-Programm nur mit zwei Floppys vom Typ 1541. Dieser Fakt gehört nun der Geschichte an. Man kann nach Lust und Laune Files zwischen allen angeschlossenen Laufwerken kopieren und verschieben. Drei kleine Mängel trüben aber die Funktionalität des Programms:

1. Die Markierung mehrer Files ist nicht möglich
2. Das Kopieren zwischen zwei Partitionen eines Speichermediums funktioniert nicht
3. Findet das Programm beim Start nicht in mindestens zwei Laufwerken Disketten, verweigert es die Arbeit und stürzt ab.

Nach Aussagen der Entwickler würde eine Erweiterung von "FileCopy" mehr Speicher benötigen – man müßte aufs Grafik-Memory zurückgreifen. In diesem Falle würden Bilddaten im 4-Bit-Speicher zerstört!

Eine erweiterte Version von "FileCopy" ist aber dennoch geplant, vor allem in Hinblick auf die Einbindung von GoDot in "64Net" (s. Kasten). Das verbesserte Programm soll dann mehrere Files markieren bzw. kopieren und zusätzliche Funktionen enthalten. Eine Belegung der Funktionstasten wäre wünschenswert. Als Vorbild könnte der "Norton Commander" (PC) oder "Directory Opus" (Amiga) dienen.

Fazit

Die neue Version von "GoDot" kann sich sehen lassen. Trotz einiger Bugs läuft das System so gut wie fehlerfrei. Wer noch keine Festplatte besitzt, benötigt die verbesserte Version von GoDot nicht unbedingt. Besitzer einer 1571 sollten schon wegen des verbesserten Moduls *"mod..FileCopy"* zuschlagen. Das Preis/Leistungs-Verhältnis läuft außer Konkurenz, da die Software als "Freeware" vertrieben wird.

Jörn-Erik Burkert

Info "64NET": Michael Renz, PP Europe, Holzweg 12, 53332 Bornheim, Tel./Fax: 02227/3221

64NET und GoDot

Im Bestreben GoDot weiter auszubauen, haben die Entwickler Arndt Dettke und Wolfgang Kling jetzt auch unter GoDot den PC zum Sklaven des C 64 gemacht. Die von Paul Gardner-Stephen (Australien) entwickelte Software, wird momentan an GoDot angepaßt und befindet sich in der Testphase.



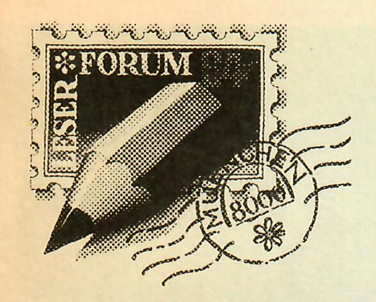
"NetCopy": Ein Modul für das "64Net" unter GoDot. Es zeigt ein PC-Directory und einige getaggte Files kurz vor der Übertragung – Zieldrive ist Unit 10 (links oben), der Quellpfad wird darunter eingegeben (Path).

Wo gibt's die Software?

Da das Software-Paket erst kurz vor Redaktionsschluß bei uns eintrudelte, konnten wir die Programme nicht rechtzeitig auf die Masterdisk zum Heft packen.

Alle interessierten Leser finden deshalb das GoDot-Update im nächsten Monat auf der Diskette zum Heft. Wer nicht so lange warten kann und ein Modem besitzt, kann sich die Software per BTX (Datex-J) oder Mailbox per Telefonkabel ziehen.

BTX: Brotkasten-Corner *matting#
Mailbox: Dream-Beam



Vizawrite-Dateien sichern

Problem von Peter Paul Gauglitz, veröffentlicht in der 64'er 4/95: Beim Aktivieren der Ladefunktion (Menüpunkt 1) landen in letzter Zeit immmer häufiger andere Textdateien im Computer-RAM als gewünscht. Meist sind das nachbearbeitete Files, die unter demselben Namen nochmals gespeichert wurden.

Nach meinen Erfahrungen mit Vizawrite verwendet diese Textverarbeitung beim Zurückschreiben bestehender Dateien auf Diskette den berüchtigten Replace-Befehl mit dem Klammeraffen <@. Dabei weiß doch jeder Besitzer eines älteren Modells der Floppy 1541, daß die Anweisung:

```
SAVE"@:Filename",8
```

in ungünstigen Fällen ein Chaos anrichtet. Deshalb ist unbedingt darauf zu achten: Vor dem Sichern des geänderten Textes ist die alte Datei zu löschen!

Allerdings tritt das Problem von vorneherein nicht auf, wenn man eine kleine Änderung in Vizawrite einbaut:

1. Laden Sie das Hauptprogramm von Vizawrite mit einem Maschinensprache-Monitor.
2. Suchen Sie die Zeichenkombination "@:" und ändern sie diese in "0:".
3. Schreiben Sie anschließend das geänderte Vizawrite-Hauptprogramm auf Diskette zurück.

Versucht man künftig, bestehende Dateien zu überschreiben, erhält man eine Fehlermeldung (Datei existiert). Man ist also gezwungen, das Text-File auf Disk vorher zu löschen oder es unter anderem Namen zu speichern.

Für die fehlerhafte Daten-Diskvon P.P. Gauglitz gibt's nur eine Rettung: nacheinander jeden Text mit Vizawrite laden und auf einer neuen Disk ablegen. Dort wird er mit dem vorgesehenen File-Namen gesichert

Andreas Rathke, Kamenz

Perpetuum mobile

Meine Floppy gibt unmittelbar nach dem Einschalten Geräusche von sich, als würde sie z.B. ein Spiel laden - obwohl gar keine Disk im Laufwerk liegt. Auch wenn ich das DIN-Kabel zum Computer entferne, oder den C 64 aus- und einschalten, ändert sich nichts. Was kann ich tun, um meine Diskettenstation wieder flott zu machen? Wo gibt es Computerfachhändler, die solche Laufwerke anbieten?

Mattias Korn, Kieselbronn

Wer kennt das Problem und weiß, wie man Matthias helfen kann? Natürlich bleibt grundsätzlich noch immer der Weg zum autorisierten Fachhändler.

Red. 64'er

Temperatursteuerung mit dem C 64

Ich habe vor, einen meiner C 64 zur Steuerung einer Anlage für solare BrauchwasserErhitzung einzusetzen. Dazu fehlen mir allerdings entsprechende Bauanleitungen oder Hinweise auf Lieferanten, die Bausätze für Temperaturregler herstellen. *Dipl.-Ing. Gerhard Paulus, Bad Kreuznach*

Wer weiß Rat?

Plagiat?

Kürzlich bin ich beim Blättern in älteren C-64-Magazin auf etwas Merkwürdiges gestoßen: In der 64'er 10/94, Seite 46, wurde ein universelles Kopierprogramm vorgestellt ("Maverick V5.04"), das man jetzt – angeblich brandneu – auch in Deutschland bekommt. Ich allerdings besitze das Programm schon seit zwei Jahren: "Renegade V 2.04". Der einzige Unterschied besteht offensichtlich in der Anzahl der Disketten: Die Renegade-Version braucht lediglich eine einzige, doppelseitig bespielte Disk. Ansonsten scheint es aber dasselbe Programm zu sein. Ist eine Raubkopie der von Ihnen vorgestellten Software "Maverick V5.04"?

Andreas Pätzold, Halle

Beide Software-Produkte haben zweifellos Gemeinsmkeiten, doch Renegade V2.04 ist nur der bedeutend unkomfortablere Vorgänger von Maverick V5.04. Die jetzt gültige Fassung wurde um etliche Funktionen erweitert, außerdem hat man diverse Bugs der Renegade-Version ausgemerzt. Wir empfehlen, sich auf alle Fälle die neue Fassung "Maverick V5.04" zuzulegen.

Red. 64'er

Public-Domain-Grafikprogramm

Die PD-Software "PGM-BASIC" bietet phantastische Möglichkeiten zur Programmierung von Grafik. Kann ich das Programm in eigene Software-Entwicklungen einbauen und diese dennoch einem Verlag zur Veröffentlichung anbieten? Muß im Programmvorspann ein Hinweis auf den Software-Entwickler oder den jeweiligen PD-Versender stehen?

Rene Kräußlich, Sonneberg

Wenn Sie den gesamten Programm-Code unverändert in eigene Software einbauen wollen, brauchen Sie dazu das Einverständnis des Autors (das ist bei PD-Software nicht anders als bei kommerziellen Programmen). Sind es nur einzelne Unterroutinen oder Programmteile, gibt es keine Hindernisse – dennoch gebietet es die Fairneß, einen entsprechenden Autorenhinweis in Ihrer neuen Kreation unterzubringen.

Red. 64'er

Richtige Verbindung zum Drucker

Problem von Werner Engler, nachzulesen in der 64'er 4/95: Mein Star LC-20 bringt mit Geos 2.0/2.5 völlig unbefriedigende Ergebnisse, wenn ich das Wiesemann-Interface 92000 7 und den Treiber "!!SP 180 VC" verwende!

Am besten benutzt man ein simples Druckerkabel und das Treiberprogramm "NLQ-Spezial" (auf der Druckertreiber-Disk zu Geos 2.5).

Die korrekten DIP-Schalter-Positionen: 1-1, 1-4, 1-5, 1-7, 2-1, 2-2 und 2-4 müssen auf ON stehen.

Steffen Brumm, Berlin

Lichtschranke nicht relevant

Frage von Martin Marggraf, abgedruckt in der 64'er 4/95: Wie programmiert man Abfragen in professioneller Software, die nach der Aufforderung "Diskette wenden!" wie von Geisterhand weitermacht, wenn die Scheibe gewechselt wird? Hat das etwas mit der Lichtschranke zu tun? Wie kann das Programm erkennen, daß die Scheibe ausgetauscht wurde?

Sobald man eine Disk nach der genannten Meldung aus dem Laufwerk nimmt, macht das Floppysystem einen Leseversuch nach dem anderen (auch, wenn gar keine Disk im Schacht liegt). Das Programm läuft also unverdrossen weiter, bis es auf eine Diskette stößt (und damit Daten liest). Die Entfernung der Disk wird durch dasselbe Programmsystem erkannt - mit der Lichtschranke der Floppystation hat das nichts zu tun. Damit werden Diskettenwechsel lediglich registriert, um zu verhindern, daß unterschiedliche Scheiben mit ein- und demselben Directory beschrieben werden.

Marius Wecker, Bad Reichenhall

Z80-Prozessor im C 128

In meinem C-128D (Plastik) habe ich einen Chip mit 4 MHz Taktfrequenz entdeckt: den Z80 von Zilog. Wie kann man ihn aktivieren (z.B. in Basic oder Assembler), um neben 1 und 2 MHz auch den Super-Speed von 4 MHz zu nutzen?

Carsten Böhle, Paderborn

Es ist keine Frage des entsprechenden Programm-Codes, den Z80 einzuschalten – es liegt vielmehr am jeweiligen Betriebssystem des C 128: der Z80 läßt sich nur mit CP/M-Plus betreiben – fürs normale C-128-System (mit dem also Basic- und Assembler-Programme arbeiten) ist der 8502-Mikrochip zuständig. Bedingt durch die im C 128 integrierte Hardware lassen sich niemals beide CPUs zusammenschalten oder gleichzeitig nutzen. Lediglich beim Einschalten des C 128 überprüft der Computer per CPU 8502, ob eine CP/M-Systemdisk im Laufwerk liegt und schaltet dann automatisch zum Z80 um: CP/M wird geladen und ist ab sofort aktiv. Vergleichbar mit MS-DOS akzeptiert es residente und externe Befehle wie DIR, FORMAT, DEL etc. Der Zeilen-Editor (ED) erlaubt es, Batch-Dateien zu erzeugen, ansonsten ist keine Programmsprache eingebaut. Allerdings existieren unter CP/M Basic-Dialekte, z.B. MBASIC, und entsprechende Makro-Assembler für Z80-Maschinensprache.

Falls Sie an weiteren Infos interessiert sind, wenden Sie sich bitte an: Helmut Jungkunz, Zacherlstr. 14, 85737 Ismaning.

Red. 64'er

Hinweis: Sowie Leser uns Problemlösungen zusenden, werden diese individuell an den Fragesteller weitergeleitet. Die Veröffentlichung zu Gunsten aller Leser folgt im nächst erreichbaren Heft. *Die Red.*

SIE KOMMT ZU IHNEN INS HAUS AM 21.7.95

Schwerpunkt: Spiele

In der August-Ausgabe dreht sich alles um C-64-Games

Spiele auf Disk

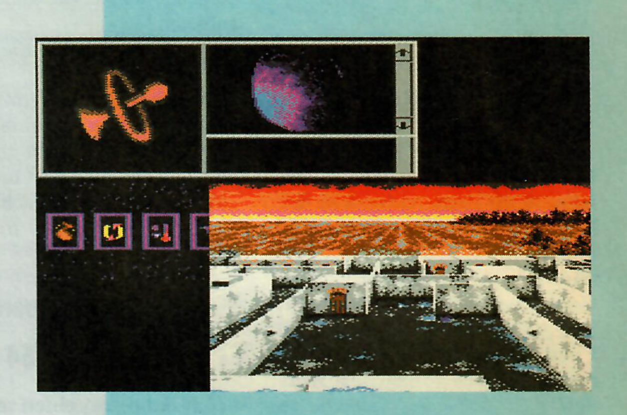
Spaß am Joystick garantiert unser Spiele-Cocktail. Einfach die Programme von Diskette laden – und schon kann's losgehen!

Tips und Tricks

Die Spiele-Hotline im 64'er-Magazin: Karten, POKEs und Kniffe auf einen Blick! Nie wieder Probleme mit Spielen!

Spiele-Previews

Wir werfen einen Blick auf kommende Spiele-Hits und bieten exklusiv Demos auf Diskette zum Probespielen!



Hardware-Test:

Kein Spiel ohne Eingabegerät! Wir nehmen das neue "Tecno-Plus-Joypad" unter die Lupe und testen den Preisbrecher.



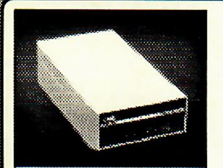
Archivatoren: Programme und Files – gut aufgeräumt!

Packen Sie mit uns Einzeldateien auf Disk komfortabel und platzsparend in ein File-Archiv! Wir zeigen Ihnen alle bekannten Lösungen von A wie Arcer bis Z wie Zipper! Dazu unser Programm-Service auf der Diskette zum Heft.

Inserentenverzeichnis

CMD 51
Data House 2
Discount 2000 29
Elektronik-Technik 23
Geos-User-Club 39
Mükra Datentechnik 33
R2/B2 ComService 5
Renz 25
Scanntronik 52
Stonysoft 29
WAW-Elektronik 29

CMD - Das Höchste in Sache C64/128-Leistung

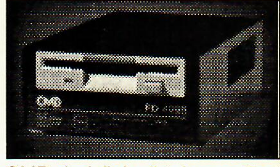


HD Series
HARD DRIVES

Das letzte Wort im Groß-Speicher für den C-64/128

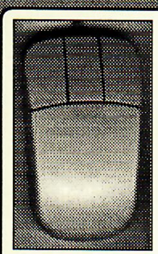
Hier haben Sie Großspeicherfestplatten für Ihren C-64 und C-128. Sofort in Betrieb zu nehmen mit leichter Installation via Seriellport. Die HDs werden mit einer Echtzeituhr, Dienstprogrammen, Kopierprogrammen, und deutschem Handbuch geliefert. Voll kompatibel mit GEOS, CP/M, BBS und vielen anderen Applikationen. HD Series-Festplatten sind in 20 MB, 40 MB, 170 MB, 270 MB, 500 MB, 1 GB Kapazitäten erhältlich.





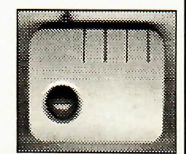
Großkapazitätslaufwerke für 1,6- und 3,2- MB-Disketten

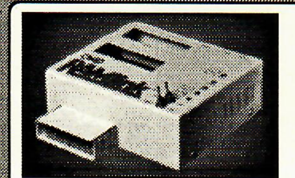
Die leistungsstarken neuen 3,5" Laufwerke von CMD sind rückwärtskompatibel mit Commodore's 1581 (800K)Laufwerk. Diese Laufwerke tun aber viel mehr! Beide Modelle haben einen 1,6MB Speichermodus und die FD-4000 bietet zusätzlich einen 3,2MB-Modus an. Beide Geräte sind JiffyDOS kompatibel (LaufwerksROM eingebaut) und haben eine Echtzeituhroption. Die FD-Series ist perfekt für GEOS, BBS-Betrieb und vieles mehr.



CMD-SmartMaus und SmartTrackball

Kein 1351-Klone! Es handelt sich hier um proportionale Eingabegeräte, die viel mehr können! Die Doppelklick-Funktion ist selbstverständlich dabei und eine Echtzeituhr für automatisches Setzen der GEOS-Uhr ist eingebaut. Beide Geräte sind mit der 1351 voll kompatibel und daher für andere Programme geeignet. Akkugepuffert und mit einer Utilities-Diskette und deutscher Anleitung geliefert. Sie sind zweifellos die Eingabegeräte der Gegenwart und Zukunft.





RAMLink

Akkubetriebenes und erweiterbares RAM-Laufwerk

RAMLink gibt Ihrem Computer ein schnelles RAM-basiertes Laufwerk, das Programme und Daten auch nach Ausschalten des Computers speichert. Der als Option angebotene Akku bietet Ihren Daten bei Stromausfall Sicherheit. Die RAMLink läßt sich auf 16MB erweitern, hat einen Pass-Thru-Port für Modulgebrauch und einen RAM-Port für den Gebrauch mit vorhandenen RAM-Erweiterungen (REU oder GEORAM). Ein Parallelanschluß ermöglicht Hochgeschwindigkeitsdatenübertragungen von und zu CMD's HD. Eingebaute JiffyDOS-Rountinen erleichtern Diskettenzugriffe und bescheunigen den Laufwerksbetrieb mit Systemen, die mit JiffyDOS ausgestattet sind. Einmalig in der Zusammenarbeit mit GEOS, Textverarbeitungen, Telekommunikationssoftware und mehr.

Preisliste

RAMLink	
RAMLink 1 MB mit Echtzeituhr	DM 465,-
RAMLink 4 MB mit Echtzeituhr	DM 699,-
RAMLink Puffer-Batterie	DM 70,-
Parallelkabel	DM 35,-
1MB/4MB SIMM	DM 89/299,-
HD-Serie Festplatten	
HD-20, 20 MB SCSI-Festplatte	DM 599,-
HD-40, 42 MB SCSI-Festplatte	DM 750,-
HD-100, 170 MB SCSI-Festplatte	DM 899,-
HD-200, 270 MB SCSI-Festplatte	DM 999,-
HD-500, 500 MB SCSI-Festplatte	DM 1199,-
HD-1000, 1 GB SCSI-Festplatte	DM 1799,-
FD-Serie Laufwerke	
FD-2000 Großspeicherlaufwerk (1,6MB)	DM 399,-
FD-2000 mit Echtzeituhr	DM 425,-
FD-4000 Großspeicherlaufwerk (3,2MB)	DM 529,-
FD-4000 mit Echtzeituhr	DM 555,-
10er-Packung, HD-Disk. (1,6 MB)	DM 35,-
10er-Packung, ED-Disk. (3,2 MB)	DM 85,-
JiffyDOS* (Bitte Pinzahl angeben C64)	
C64-System (24 oder 28 Pin Kernal)	DM 75,-
SX-64-System	DM 75,-
C-128-System	DM 99,-
128-D-System (Metallgehäuse)	DM 99,-
128-D-Tragbar (Kunststoffgehäuse)	DM 99,-
Zusätzliche Laufwerke-ROM's	DM 40,-
Weitere Soft- u. Hardware	
CMD-SmartMaus	DM 99,-
CMD-SmartTrackball	DM129,-
GEOS Paralleldruckerkabel	DM 59,-
EX3 Modulportweiche	DM 69,-
SwiftLink-RS232	DM 99,-
SID Symphony Stereo Cartridge	DM 99,-
Big Blue Reader	DM 55,-
Compression Kit'94	DM 75,-
Dialogue 128	DM 50,-
geoMakeBoot	DM 22,-
gateWay/64 oder gateWay/128	DM 60,-
geoProgrammer (ML für GEOS)	DM 99,-
GEOBasic (BASIC für GEOS)	DM 45,-
Collette Utilities	DM 60,-
GEOS Power Pak I o. II/Companion je	DM 45,-
IPAINT	DM 79,-
IPORT	DM 65,-
GEOS & GEOS Applications (US)	Anrufen



Andere CMD Produkte

EX3 Modulportweiche

Die neue EX3-Weiche von CMD ist bestückt mit neuen Features für den modernen Commodore-Power-User. Schauen Sie sich diese Features an:

- Jeder Port ermöglicht separates Schalten von +5, GAME, EXROM, ROMH, ROML, IO1, und IO2 Leitungen sowohl das gänzliche Ausschalten des Ports.
- Der Mittelport besitzt einen IO1/IO2 ($DE00/$DF00) Swapschalter.
- Reset-Schalter

Die EX3 ist für den Gebrauch mit SwiftLink, SID Symphony, REUs, GEORAM und Utility-Modulen geeignet.

SID Symphony Stereo Modul



SID Symphony verdoppelt die Soundstimmen Ihres C64ers/C128ers. SID Symphony kann mit dem Ausgang Ihres Computers kombinert oder an Ihr Stereogerät angeschlossen werden. Der SID Stereo Player und Demo-Songs werden mitgeliefert.

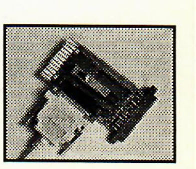


SwiftLink-RS232 Modul

SwiftLink bietet dem C64/C128 das an, was andere Computer seit Jahren besitzen: einen echten RS-232 Port. SwiftLink unterstützt Datenübertragungsgeschwindigkeiten von bis zu 38,400 BAUD und ermöglicht das Anschließen moderner Hochgeschwindigkeitsmodems an den Commodore. SwiftLink kann auch als Nullmodem mit anderen Computern für schnelle Datenübertragungen benützt werden. SwiftLink wird mit DFÜ-Software und Dateitransfersoftware geliefert, um das Konvertieren von PET ASCII auf Standard-ASCII zu ermöglichen. Voll kompatibel zum neuen GEOFAX-Programm.

GEOCABLE II Paralleldruckerkabel

Das fortgeschrittenste Paralleldruckerkabel, was je gebaut wurde! Diese durchgeschliffene Userport-Weiche besitzt einen Pass-Thru-Switch, um zwischen Drucker und einem anderen angeschlossenen Gerät ohne Konflikt zu schalten. GEOS-Treiber und Utilities werden mitgeliefert. Voll kompatibel mit u.a. GEOS, Superbase, SuperScript, GEOS-LQ, Action-Replay MK VI.



* Es ist wichtig, Ihre Computer- und Laufwerksmodellnummer bei der Bestellung anzugeben. C64-Besitzer müssen die Seriennummer auch angeben. Ein JiffyDOS-System beinhaltet die ICs für einen Computer und ein Diskettenlaufwerk.

Preise enthalten Fracht, Zoll und Steuer. Lieferung ca. 4 bis 6 Wochen. Preise können ohne Vorankündigung geändert werden. Bei Vorkasse keine Versandkosten. NN + 10,00 DM

CMD Direkt
CMD Direkt Sales
Postfach 58
A-6410 Telfs, Austria
Tel.: 0043-5262-66080
CompuServe: 100423,2717